EPFL

Thèse n° 8654

Representation Learning for Multi-relational Data

Présentée le 3 décembre 2021

Faculté des sciences et techniques de l'ingénieur
Programme doctoral en génie électrique

pour l'obtention du grade de Docteur ès Sciences

par

Eda BAYRAM

Acceptée sur proposition du jury

Prof. D. N. A. Van De Ville, président du jury
Prof. P. Vandergheynst, directeur de thèse
Prof. X. Dong, rapporteur
Prof. E. Vural, rapporteuse
Prof. N. Kiyavash, rapporteuse

École
polytechnique
fédérale
de Lausanne

2021

We would like to love more, laugh more, think more.
We would like to see more, understand more, trust more.
All we have left now are these wishes and these words.
There was peace, dreams, sleep and kisses.
There were also people, fruit, paper and pens.
After all, nothing can be as astounding as life.
Except for writing. Yes, of course, except for writing,
the sole consolation.
— Orhan Pamuk

Abstract

Recent years have witnessed a rise in real-world data captured with rich structural information that can be better depicted by multi-relational or heterogeneous graphs. However, research on relational representation learning has so far mostly focused on the problems arising in simple, homogeneous graphs. Integrating the structural priors provided by multi-relational data may further empower the generalization capacity of representation learning models, yet it still remains an open challenge. Although there is a strong line of works on relational machine learning on knowledge graphs, it is quite concentrated on the task of completing missing edges, which is known as link prediction. In this thesis, we shift the focus away from the well-addressed node and graph classification problems on simple graphs or the link prediction problem on knowledge graphs, and prompt new research questions targeting the representation learning problems that are overlooked in multi-relational data.

First, we focus on the problem of node regression on multi-relational graphs, noting that inference of continuous node features across a graph is rather under-studied in the current relational learning research. We propose a novel propagation method which aims to complete missing features at the nodes of a multi-relational and directed graph. Our multi-relational propagation algorithm is composed of iterative neighborhood aggregations which originate from a relational local generative model. Our findings show the benefit of exploiting the inductive bias led by the multi-relational structure of the data.

Next, we consider the node attribute completion problem in knowledge graphs, which is relatively unexplored by the knowledge graph reasoning literature. We propose a novel multi-relational attribute propagation method where we harness not only the relational structure of the knowledge graph, but also the dependencies between various types of numerical node attributes relying on a heterogeneous feature space. Our algorithm is framed within a message-passing scheme where the propagation parameters are estimated in advance. We also propose an alternative semi-supervised learning framework where the parameters and the missing node attributes are inferred in an end-to-end fashion. Experimental results on well-known knowledge graph datasets relay the effectiveness of our message-passing approach, which specifies the computational graph by the heterogeneity of the data.

Finally, we study graph learning in multi-relational data domain. Unlike the existing structure inference methods, we aim at exploiting and combining each source of relational information

provided by the data domain to learn the underlying graph of a set of observations. For this purpose, we employ a multi-layer graph representation which encodes multiple types of relationships between data entities. Then, we propose a mask learning method to infer a specific combination of the layers which reveals the structure of observations. Experiments conducted both on simulated and real-world data suggest that incorporating multi-relational domain knowledge enhances structure inference by boosting its adaptability to a variety of input data conditions.

Key words: multi-relational data, relational representation learning, knowledge graph reasoning, node attribute completion, multi-relational propagation, heterogeneous graphs, heterogeneous node regression, message-passing, structure inference, graph learning

Résumé

Ces dernières années, la quantité de données contenant des informations structurelles riches a rapidement augmenté. Ces données peuvent être représentées par des graphes relations multiples ou hétérogènes. Cependant, les recherches sur l'apprentissage de représentations relationnelles se sont jusqu'à présent principalement concentrées sur les problèmes posés par des graphes simples et homogènes. L'intégration des a priori structurels fournis par les données multi-relationnelles peut renforcer la capacité de généralisation des modèles d'apprentissage, mais cela reste un défi ouvert. Bien que de nombreux travaux portent sur l'apprentissage machine sur les graphes de connaissances, ces travaux se sont principalement concentrés sur la complétion d'arêtes manquantes, ce qui est connu sous le nom de prédiction de lien. Dans cette thèse, au lieu de concentrer notre attention les problèmes bien traités de classification de nœuds et de graphes sur des graphes simples ou du problème de prédiction de liens sur des graphes de connaissances, nous posons de nouvelles questions de recherche ciblant les problèmes négligés d'apprentissage avec des données multi-relationnelles.

Tout d'abord, nous nous concentrons sur le problème de la régression des nœuds sur les graphes multi-relationnels, notant que l'inférence des attributs de nœuds continus à travers un graphe est plutôt sous-étudiée dans la recherche actuelle sur l'apprentissage relationnel. Nous proposons une nouvelle méthode de propagation qui vise à compléter les attributs manquants aux nœuds d'un graphe multi-relationnel et orienté. Notre algorithme de propagation multi-relationnelle est composé d'agrégations de voisinage itératives qui proviennent d'un modèle génératif local relationnel. Nos résultats montrent l'intérêt d'exploiter le biais induit par la structure multi-relationnelle des données.

Ensuite, nous considérons le problème de complétion des attributs de nœuds dans les graphes de connaissances, qui est relativement peu exploré par la littérature. Nous proposons une nouvelle méthode de propagation multi-relationelle d'attributs où nous exploitons non seulement la structure relationnelle des graphes de connaissances, mais aussi les dépendances entre divers types d'attributs numériques de nœuds reposant sur un espace de caractéristiques hétérogène. Notre algorithme est utilisé dans un schéma de transmission de message où les paramètres de propagation sont estimés à l'avance. Nous proposons également un cadre d'apprentissage semi-supervisé alternatif où les paramètres et les attributs de nœuds manquants sont inférés de bout en bout. Les résultats expérimentaux sur des ensembles de graphes de connaissances bien connus montrent l'efficacité de notre approche de transmission de

messages, qui spécifie le graphe de calcul par l'hétérogénéité des données.
Enfin, nous étudions l'apprentissage de graphes dans le domaine des données multirelationnelles. Contrairement aux méthodes d'inférence de structure existantes, nous visons à exploiter et à combiner chaque source d'informations relationnelles fournies par le domaine de données pour apprendre le graphe sous-jacent d'un ensemble d'observations. À cette fin, nous utilisons une représentation graphique multi-couche qui code plusieurs types de relations entre les entités de données. Ensuite, nous proposons une méthode d'apprentissage par masque pour déduire une combinaison spécifique des couches qui révèle la structure des observations. Des expériences menées à la fois sur des données simulées et réelles suggèrent que l'incorporation de connaissances multi-relationnelles améliore l'inférence de structure en augmentant son adaptabilité à une variété de conditions de données d'entrée.

Mots clefs: données multi-relationnelles, apprentissage de représentations relationnelles, graphes de connaissances, complétion d'attribut de nœud, propagation multi-relationnelle, graphes hétérogènes, régression de nœud hétérogène, transmission de message, inférence de structure, apprentissage de graphes

Contents

Abstract (English/Français) i

List of figures vii

List of tables ix

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Thesis Outline . . . 4

2 Overview of Relational Learning from the Perspective of Smoothness 7
- 2.1 Learning Representations for Graph-Structured Data . . . 8
  - 2.1.1 Graph Regularization . . . 9
  - 2.1.2 Neighborhood Aggregation . . . 11
- 2.2 Inverse Problem: Inference of the Underlying Graph Structure . . . 18
  - 2.2.1 Inverse Covariance Estimation . . . 19
  - 2.2.2 Estimation of Graph Laplacian . . . 20

3 Multi-Relational Propagation for Node Regression 23
- 3.1 Multi-relational Model . . . 24
  - 3.1.1 First-order Relational Bayesian Estimate . . . 25
  - 3.1.2 Estimation of Relational Parameters . . . 26
- 3.2 Multi-relational Propagation Algorithm . . . 28
- 3.3 Experiments . . . 31
  - 3.3.1 Multi-relational Estimation of Weather Measurements . . . 31
  - 3.3.2 Predicting People's Date of Birth in a Social Network . . . 34
- 3.4 Conclusion . . . 36

4 Heterogeneous Message Passing in Knowledge Graphs 37
- 4.1 Completion of Numerical Node Attributes in Knowledge Graphs . . . 38
- 4.2 Multi-Relational Attribute Propagation . . . 41
  - 4.2.1 Heterogeneous Local Generative Model for Numerical Attributes . . . 41
  - 4.2.2 Algorithm MRAP . . . 44
- 4.3 Semi-supervised Learning Scheme . . . 46
- 4.4 Experiments . . . 49

4.4.1 Performance of MRAP . . . 49
4.4.2 Performance of Semi-Supervised Learning Scheme . . . 53
4.5 Conclusion . . . 56

5 Graph Learning in Multi-Relational Data Domain 57
5.1 Mask Combination of Multi-layer Graphs . . . 57
5.1.1 Comparison to the Related Learning Schemes . . . 59
5.1.2 Contributions . . . 61
5.2 Mask Learning Algorithm . . . 61
5.2.1 Multi-layer Graph Settings . . . 61
5.2.2 Mask Combination of Layers . . . 62
5.2.3 Problem Formulation . . . 63
5.2.4 Discussion . . . 64
5.3 Experiments . . . 67
5.3.1 Experiments on Synthetic Data . . . 68
5.3.2 Learning from Meteorological Data . . . 74
5.3.3 Learning from Social Network Data . . . 78
5.4 Conclusion . . . 80

6 Conclusion 83
6.1 Summary of Contributions . . . 83
6.2 Open Research Directions . . . 84

A Appendix of Chapter 2 87
A.1 Derivation of Graph Regularization Term as the Quadratic Form of Laplacian . 87
A.2 Derivation of Graph Regularization Solution . . . 88
A.3 Iterative Approximation of Graph Regularization Solution . . . 88
A.4 Negative Log-Likelihood Estimation with the Local Factor Analysis Model . . . 88

B Appendix of Chapter 3 91
B.1 Gradient of the Loss in Problem (3.3) . . . 91
B.2 Negative Log-Likelihood Estimation of the parameters of the Relational Local Generative Model . . . 91

Bibliography 103

Curriculum Vitae 105

List of Figures

1.1 Depiction of a heterogeneous biomedical network 2

3.1 A fragment of a multi-relational and directed social network 24
3.2 Distribution of change in temperature and snowfall(cm) measurements between the weather stations that are related via altitude proximity. Differences are shown along the ascend and descend direction separately, then, symmetric distribution shows the changes regardless of the direction. Also, a radial basis function (RBF) is fitted to each histogram. 32
3.3 Distribution of change in precipitation(mm) measurements between the weather stations that are related via geographical and altitude proximity. 34
3.4 Distribution of difference (year) in date of births over different types of relations between people. 34

4.1 A part of KG data with incomplete node attributes 39
4.2 Message passing performed for updating the attribute `date_of_death` for the node Francis Ford Coppola. 43
4.3 Update of a node attribute in one iteration of forward propagation. For the sake of grouping the messages, we grant that $\mathtt{r}(v,v)$ returns null. 47
4.4 A summary of FB15K-237 with entity types and numerical attributes encountered on them. The number attached to the connection between a pair of entity types indicates the number of relationship types between those entities. 49
4.5 Histograms and fitted normal curves of node attribute differences computed along some relations 50
4.6 Learning curves of SSL-1, SSL-2 without dropout, SSL-2 with dropout from left to right. 55

5.1 An illustration for the input and output of the mask learning algorithm 59
5.2 Performance with respect to the ratio of layer edges 70
5.3 Ground truth global graph and the solution given by **ML** 71
5.4 Performance of **ML** with different γ values vs coverability 72
5.5 Performance of the algorithms vs coverability 72
5.6 Performance of the algorithms vs number of signals 73
5.7 Performance of the algorithms vs signal quality 74
5.8 Year average of temperature and precipitation 76

5.9 Sparsity pattern of the layers and the masks with respect to year average of temperature 76
5.10 Performance of **ML** ($\gamma = 0.6, \Gamma = 32$) on CS-AARHUS data 79
5.11 Performance of the graph learning algorithms vs number of signals in lunch data 80

6.1 Overview of the pipeline for development of a propagation algorithm 84

List of Tables

3.1 Local Generative Model and Operation in Simple and Multi-relational Graphs . 27
3.2 Temperature and Snowfall Prediction Performances 32
3.3 Precipitation Prediction Performances 33
3.4 Statistics for each type of relation. Columns respectively: number of edges, mean and variance of the date of birth difference belonging to the associated relation type. 35
3.5 Date of Birth Prediction Performances 35

4.1 Number of node attributes encountered in datasets for each attribute type. The upper block contains numerical attributes of date type. The lower block contains all other attributes. A dash (-) indicates the corresponding attribute is not encountered in the dataset. 50
4.2 (Upper) Dataset statistics. (Lower) Characteristics of MRAP in these datasets. . 51
4.3 Performances on FB15K-237 with two different setup of observed node attribute sparsity 52
4.4 Performances on YAGO15K with two different setup of observed node attribute sparsity 52
4.5 Ablation study for MRAP. MAE measured on the experimental setup '50%'. . . . 53
4.6 Performances of different learning frameworks on YAGO15K date attributes . . 55

5.1 Global Graph Recovery and Mask Recovery Performances 70
5.2 Contribution of layers on the structure of different measurements 75
5.3 Signal inpainting performance of the algorithms 78
5.4 Performance of the methods in recovering the lunch network 79

1 Introduction

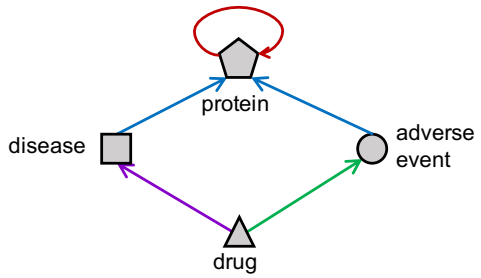
1.1 Motivation

Representation learning achieved great milestones in the last decade in image and speech recognition where the data is described by a regular structure. In many settings, data possess complex relational structures rather than sequential or grid patterns [1]. For instance, biological networks [2], molecules [3, 4, 5] and physical systems [6, 7] are often contemplated as a system of interacting elements, which is inherently represented by a graph through its nodes and edges between them. This has led to relational machine learning frameworks leveraging these structural priors such as graph representation learning [8], deep learning on graphs [9] and geometric deep learning [10].

Various disciplines are now able to capture different level of interactions between the entities of their interest, which promotes multiple types of relationships within data. This compels novel strategies to process emerging multi-relational forms of data. Examples include social networks that relate individuals based on different types of connections or behavioral similarities [11, 12], biological networks where different modes of interactions exist between neurons or brain regions [13, 14], biomedical networks that are organized in multiple types of interacting elements [15], an illustration is provided in Figure 1.1, and transportation networks which organize people's movement via different means of transportation [16, 17]. This thesis investigates new methods to solve certain representation learning problems arising in multi-relational data.

Given the rising complexity in real-world network structured data, it is also required to properly organize such diverse information in order to conduct further processing and inference tasks. The graph structures accommodating multiple nodes and edges features are superior for this purpose, rather than simple and homogeneous graphs.

Multi-relational Graphs. We now mention several graph structures which broadly allow multiple relationships within its body, and that are utilized in our work. To start with, multi-layer graph representations are convenient for encoding complex relationships of multiple types between data entities [18]. In general, each layer encodes a distinct relational context



(a) Semantic Model

protein	interacts	protein
disease	associates	protein
drug	treats	disease
drug	causes	adverse event
adverse event	associates	protein

(b) Relation types

Figure 1.1 – Depiction of a heterogeneous biomedical network

among the data entities. While multi-layer networks have gained considerable attention from the field of network science [19], they are not yet well-noticed by the machine learning community. Furthermore, heterogeneous networks permit storing rich structural information within data [20, 21]. For an illustration, in Figure 1.1, we summarize a biomedical network with a semantic model depicting different types of relations between multiple types of entities. Although, the research on relational representation learning was persistent on the problems arising in simple, homogeneous graphs, there is a rising interest in knowledge graphs (KGs), which are also counted as heterogeneous networks, consisting of multiple node and edge types. KGs play important role in a variety of AI applications including question answering [22, 23], drug discovery [24, 25], and e-commerce [26, 27].

In this thesis, we address certain representation learning problems emerged on multi-relational data, yet before we take a snapshot of the problems that are at the focus of relational learning research.

Relational Representation Learning Problems. In recent years, node and graph classification tasks have become the main focus of the research in graph representation learning. Consequently, a strong line of works has been produced for the inference of the node-level and graph-level categorical features in transductive and inductive settings [1]. Despite this, there has been little interest in regression of continuous node features across a graph, acknowledging some of the early works handling node regression under signal inpainting on graphs [28, 29]. In particular, node regression on multi-relational graphs still remains unexplored.

Moreover, edge-level inference of categorical features is substantially studied in KGs for the completion of missing connections between entities, which is referred to as link prediction. For instance, statistical relational learning [30] and KG embedding methods [31] have proposed solid frameworks for prediction of one-hop relations in multi-relational data. Then, recent query embedding methods [32, 33] enable multi-hop reasoning which can answer complex queries, such as "Which protein is associated with the adverse event caused by the drug X?", see Figure 1.1. KG reasoning studies often address the prediction of relations in incomplete KGs—containing missing facts, whereas the incompleteness in the node attributes of KGs is quite overlooked. Especially, inferring various types of categorical and continuous features

possessed by different types of entities is still an open challenge.

Relational representation learning methods inherently assume that the relational structure of the data is explicit. As this may not be always the case, the structure underlying a certain downstream task can also be implicit. When the underlying graph is latent, it can be inferred from the observations. This is achieved by some of the early works which impose a relational statistical model on the observations [34]. The statistical model, in general, prescribes connecting the nodes of a graph whose observations hold a notion of similarity, which is often referred to as smoothness. Although real-world data is often captured with a certain domain knowledge accommodating complex relationships, such background information is not well exploited by the existing structure inference methods. Particularly, how to leverage multi-relational semantics of the data to discover the structure that is specific to the task of interest is not yet well understood.

We note that in terms of reasoning, the structure inference problem follows a reverse path compared to the representation learning problem which broadly aims at obtaining similar representations at the connected nodes of a graph. Therefore, the relational structure of the data offers a rewarding inductive bias, which can improve the generalization capacity of the representation learning models [6, 35]. In this thesis, we draw attention to the feasibility of exploiting multi-relational semantics of data, which may further offer the augmentation of relational reasoning and empower the ability of abstraction in relational learning frameworks.

Challenges. Having stated our motivation to benefit from multi-relational structure of data, we acknowledge, however, that it is not straightforward to properly deal with such a complexity and harness it in the reasoning process. In general, integrating complex structural information within a relational learning scheme is an open challenge. To begin with, a direct expansion of model parameters by the volume of multi-relational information could be problematic due to possible over-fitting issues. Especially in the case of knowledge graphs, where the structure is highly heterogeneous with different types of nodes and edges, the combinations creating a relation may expand so fast that it might require additional out-of-distribution generalization strategies.

Besides the structural information, the complexity of the feature information also requires special attention. For instance, different types of nodes usually possess different types of properties that are expressed in different feature spaces. How to properly incorporate them in the learning scheme simultaneously with the graph is one of the challenges to be managed in heterogeneous structures.

Moreover, each type of relational information may play a different role for prioritizing the structure underlying a certain task or a certain set of observations, yet, there is no evident technique for combining each relational source of information for the inference task. Nonetheless, different types of relationships between data entities usually follow different affinity rules, rather than depending on a uniform notion of similarity. This suggests cultivating the inductive bias using the multi-relational semantics accordingly.

Research Questions and Contributions. Given our motivation and regarding the challenges, we list the research questions addressed in this thesis as follows:

(Q1) Examining the state-of-the-art methods accomplishing node-level regression on graphs, can we adapt them to incorporate available multi-relational information about data domain? In particular, how can we achieve node-value imputation on a multi-relational and directed graph?

(Q2) How can we improve the multi-relational node regression strategy to achieve completion of node features in a heterogeneous graph where multiple edge and node types exist? In particular, how can we predict missing numerical attributes in a knowledge graph?

(Q3) Examining the state-of-the-art structure inference methods aiming at discovering the underlying graph structure of the data, can we support the inference process with available multi-relational information about the data domain? How can we exploit and combine the multi-relational information to reach the structure underlying a set of nodal observations?

To address the first research question, we investigate a relational model preserving the intrinsic structure of the data and propose a multi-relational node regression framework [36]. Next, we develop on top of this methodology for the task of completing missing node features on heterogeneous graphs. We propose a message-passing scheme facilitating information exchange between various types of numerical attributes over the given multi-relational structure of a knowledge graph [37]. For the last, we switch gears and focus on an inverse problem: inferring the structure from a given set of nodal observations acquired in a multi-relational data domain. We propose a novel technique for capturing task-relevant connections from each layer of a multi-layer graph and combining them into a global graph underlying the observations [38]. Ultimately, the main contribution of this thesis lies in the exploitation and combination of the available multi-relational information for representation learning. This repays with better accuracy and the interpretability of the inference task by revealing the contribution of each relational source of information within data.

1.2 Thesis Outline

This dissertation is organized into four main chapters. We start with an overview of the relational learning methodologies in Chapter 2. Since both the existing solutions in the literature and ours profoundly exploit the notion of smoothness as the inductive bias in relational learning, we revisit seminal approaches from the perspective of smoothness. Accordingly, the chapter constitutes the fundamentals for the research conducted in this thesis. We give the overview by the scope of two distinct problems in relational learning: representation learning and structure inference. We begin with the former, which aims at learning representations from a given relational structure, by analyzing the graph regularization approach. Then, we proceed with well-known graph algorithms, such as label propagation, which iteratively converge to the global solution suggested by graph regularization. We show that simple neigh-

borhood aggregation operated on a given relational structure holds the basis for these methods since they employ the smoothness prior by promoting similar representations at the neighboring nodes of the graph. We provide a re-intepretation of the neighborhood aggregation from a Bayesian perspective by imposing a local generative model on the neighboring nodes. We later improve this model with multi-relational neighborhood and propose a relational local generative model in Chapter 3 and a heterogeneous local generative model in Chapter 4. We then mention notable neural network schemes relying on a similar neighborhood aggregation principle yet providing more flexible models which boost the representational power for the subsequent machine learning tasks. In the latter section, we scrutinize the structure inference methods which aim at discovering latent relational structure of the data. We emphasize that they recruit the smoothness prior by promoting a graph connecting the nodes with similar observations, therefore, structure inference can be stated as the inverse of the representation learning problem. We first mention some early works estimating the inverse covariance (precision matrix) of the data in order to understand the dependency structure within data. Then, we make a passage to studies learning the graph Laplacian matrix as an instance of the precision matrix. We finally elaborate the methods learning the graph by building smooth signal representation model on the observations, which we also adopt in our graph learning framework in multi-relational domain, in Chapter 5.

In Chapter 3, we present a multi-relational node regression framework. We take inspiration from the well-known label propagation algorithm aiming at completing categorical features across a simple, weighted graph. While the propagation of continuous node features across a graph is rather under-studied, we take a step further and propose a novel propagation algorithm aiming at completing missing features at the nodes of a multi-relational and directed graph. We follow the propagation procedure that we break down by the neighborhood aggregations derived through a simple local generative model in Chapter 2. We extend this by incorporating a multi-relational neighborhood and suggest a relational local generative model. Then, we build our multi-relational propagation algorithm by iterative neighborhood aggregation steps originating from this new model. We provide the derivation of the parameters of relational local generative model, which can be estimated over the observed set of node features and assigned as the parameters of the proposed propagation algorithm. We compare our multi-relational propagation method against the standard propagation in several node regression scenarios. In each case, our approach enhances the results considerably by integrating the multi-relational structure of the data into the regression framework.

Next, in Chapter 4 we study the problem of numerical node attribute completion in knowledge graphs. Since knowledge graphs consist of multiple types of entities connected via different types of relationships, we extend our multi-relational propagation approach in order to impute missing heterogeneous features possessed by the entities of a knowledge graph. To this end, we introduce a heterogeneous local generative model conforming the relationship between different types of node features that are attributed to neighboring nodes or to the same node. We propose a multi-relational attribute propagation method which iteratively aggregates the node attributes based on such a model. We employ a set of message functions facilitating the

information exchange between different types of source and target attributes through different types of relations. We then frame the proposed method within a message-passing scheme where the propagation parameters are estimated in advance. We also propose an alternative semi-supervised learning framework where the parameters and the missing node attributes are inferred in an end-to-end fashion. We compare the proposed frameworks against several baseline approaches and demonstrate their effectiveness by their performance to complete numerical features in two knowledge graph datasets.

Finally in Chapter 5, we focus on the structure inference problem in multi-relational data domain. We exploit the smooth signal representation model in order to learn the graph underlying a set of observations while we integrate the available multi-relational information given by the data domain in the inference process. We employ a multi-layer graph representation, where each layer encodes one type of relational information between the data entities. Then, we propose a mask combination method, which captures and fuses relevant information from each layer specific to the structure of the observations. We show that the proposed structure inference framework is more advantageous than the state-of-the-art solutions especially when there is a limited number of observations deviating from the assumed statistical model. Incorporating the multi-relational domain knowledge, our approach not only increases the accuracy of the solution but also enables revealing the contribution of each source of relational information within data.

2 Overview of Relational Learning from the Perspective of Smoothness

This chapter draws a picture of the machine learning techniques developed for relational data by introducing the fundamentals and preliminaries; thus, it provides a base for the following chapters. Throughout the chapter, we explain how different approaches incorporate the relational structure of the data in their reasoning process, and we propose the concept of smoothness as a unifying perspective. Smoothness is a prior imposed on the representations, and the machine learning studies on graphs extensively exploited it as the relational inductive bias [6]. In the literature, graphs are conveniently used to encode relational data. A graph is denoted by $\mathcal{G}(\mathcal{V},\mathcal{E})$ where each vertex/node in $\mathcal{V}$ stands for a datum and each edge in $\mathcal{E}$ stands for a pairwise relationship within data. With respect to this notation, "smooth" node representations can be discerned by exhibiting "minimal" variations over the edges of the graph.

The chapter is separated into two sections. In the former, we concentrate on learning representations for graph-structured data. The representation learning problem can be described as learning a function capturing a certain representation of the given graph structure,

$$f : (\mathcal{V},\mathcal{E}) \rightarrow \mathbf{X},$$

where $\mathbf{X} = [x_1 \dots x_N]^\top$ is a matrix which stores the representation vectors for each node on the graph, $|\mathcal{V}| = N$. Inherently, the solution is designed with respect to a downstream task where the inferred representation is to be used. Usually, the problems on graph-structured data emerge from node, edge or graph level prediction tasks, such as node regression, link prediction, graph classification etc. We scan through the early and the recent approaches addressing relational representation learning problem by taking the smoothness assumption as the common ground.

In the latter section, we concentrate on learning the underlying graph structure of the data. This problem can be considered as the inverse of representation learning. The latent relational

structure is recovered from a set of observations:

$$f : (\mathcal{V}, \mathbf{X}) \rightarrow \mathcal{E}.$$

In this case, matrix $\mathbf{X}$ stores a given set of node features, on which the smoothness prior is imposed. A desired property for the solution, $\mathcal{E}$, relates to the sparsity of the graph structure. This is because the end goal of graph learning is usually to obtain a topological summary of the data. Such knowledge can be leveraged for capturing similarities, interactions or dependencies within data or for the subsequent prediction tasks as well.

Ultimately, the relational inductive bias can be popularized as follows. In the representation learning case, smoothness is imposed as "Nearby nodes on the relational structure should have similar representations.", whereas in the graph learning case, this is rephrased by "Nodes constituting similar representations should be neighbors."

2.1 Learning Representations for Graph-Structured Data

In this section, we focus on representation learning on graphs by traversing from the early to the recent approaches by reviewing how they handle the smoothness prior in the learning process. We first elaborate the graph regularization approach [39, 40], which revives the smoothness prior by employing the graph structure as a regularization term in the optimization problem for learning the representations. Nonetheless, computing the global optimum of such problems can be too expensive in complex data settings and learning schemes that are recently emerging. Accordingly, we mention latter approaches that aim at approximating the solution via iterative algorithms. These techniques apply sequential neighborhood aggregations on the relational structure, which is tractable, computationally cheaper, and shown to converge to the global optimum.

Smoothness prior imposes similar representations on the neighboring nodes of the graph. In this sense, the learned representations—also called embeddings— are supposed to preserve the pairwise distances of the nodes on the graph. In other words, the nodes that are close on the intrinsic relational structure will be as close as possible on the embedding space. Since such a representation would signify the global position of a node on the graph, it is also referred to as position-aware embedding [41].

In order to establish a background for the following chapters, we exemplify the problems aiming at learning node embeddings in transductive setting [42, 43], where the graph structure is fixed and known. Generalizations to inductive setting is possible in principle, yet omitted here for the sake of simplicity. We also acknowledge the line of works concentrated on unsupervised graph representation learning via matrix factorization [44, 45, 46]. Nonetheless, in this section we scrutinize the graph regularization approach and then iterative neighborhood aggregation methods. Building upon the exploitation of smoothness, some other variations and extensions can also be found in the literature. A broader taxonomy of graph representation learning

methods with a generalization from the encoder-decoder perspective is given in [1].

2.1.1 Graph Regularization

The smoothness prior is first used in [47] in order to obtain locality preserving embedding of a graph. Then, it is designated as the global consistency assumption in [48] and affirmed that the solution of a transductive learning problem is sufficiently smooth if nearby points on the relational structure are likely to have the same label. The graph regularization technique employ the graph structure as a regularizer in the optimization problem in order to obtain a smooth solution with respect to the underlying graph structure. Therefore, we can frame the objective of this problem as minimizing a loss such as:

$$\mathcal{L} = \mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{sup}}, \tag{2.1}$$

where the first term is the regularization loss exploiting the graph, and the second term is the loss supervised by a downstream task, such as node-level regression. The regularization loss measures the smoothness of the representation on the underlying graph and formulated as the sum of the local variations over the relational structure. Thus, it can simply be computed by summing up the pairwise distances between the neighboring node embeddings:

$$\mathcal{L}_{\text{reg}} = \frac{1}{2} \sum_{i \in \mathcal{V}} \sum_{(i,j) \in \mathcal{E}} \mathrm{d}(x_i, x_j), \tag{2.2}$$

where x_i is the embedding vector for node-i and $\mathrm{d}(\cdot,\cdot)$ is a kernel function measuring the pairwise distances on the embedding space. In the literature, squared Euclidean distance is commonly used for measuring the similarity between two embedding vectors. Accordingly, it is possible to express the objective by employing ℓ_2 norm distance.
Problem 1: Graph regularization with ℓ_2 sense smoothness

$$\min_{\mathbf{X}} \frac{1}{2} \sum_{i \in \mathcal{V}} \sum_{(i,j) \in \mathcal{E}} \|x_i - x_j\|_2^2 + \gamma \sum_{i \in \mathcal{V}} \|x_i - y_i\|_2^2, \tag{2.3}$$

where $\mathbf{X}$ is the representation matrix, and $\mathbf{Y} = [y_1 \dots y_N]^\top$ contains given set of node feature vectors. The first term, $\mathcal{L}_{\text{reg}}$, measures the smoothness in ℓ_2 sense, whereas the second term measures the closeness of the learned representations to the initial node features $\mathbf{Y}$, and the trade-off between them is adjusted by a hyperparameter $\gamma > 0$. Given the adjacency matrix $\mathbf{A}$ enclosing the relational structure of the graph, one can write the graph regularization term via matrix notation as follows:

$$\mathcal{L}_{\text{reg}} = \mathrm{tr}(\mathbf{X}^\top (\mathbf{D} - \mathbf{A})\mathbf{X}), \tag{2.4}$$

where $\mathrm{tr}(\cdot)$ is the trace operator and $\mathbf{D}$ is the diagonal degree matrix. The derivation of (2.4) can be found in Appendix A. Here, we designate the *graph Laplacian* matrix as $\mathbf{L} = \mathbf{D} - \mathbf{A}$ and the quadratic Laplacian form, $\mathrm{tr}(\mathbf{X}^\top \mathbf{L}\mathbf{X})$, as the *Dirichlet energy* of the representations over the

given graph structure.

The solution of Problem 1 satisfies $\frac{\partial \mathcal{L}}{\partial \mathbf{X}}(\mathbf{X}^*) = 0$, which leads to (see Appendix A for derivation):

$$\mathbf{X}^* = \gamma(\mathbf{L} + \gamma \mathbf{I}_N)^{-1}\mathbf{Y}, \tag{2.5}$$

where $\mathbf{I}_N$ is identity matrix of size N.

Now, we revisit the graph regularization framework studied by Zhou and Schölkopf [39], which shows that it is possible to converge to the solution proposed by graph regularization problem by iterative operations on the graph. In order to conduct further analysis on the global optimum proposed for Problem 1, we proceed with certain modifications on the variables: $\mathbf{L} = \mathbf{I} - \mathbf{S}$, and $\gamma = \frac{1}{\xi} - 1$, then rephrase the solution as

$$\mathbf{X}^* = (1-\xi)(\mathbf{I} - \xi \mathbf{S})^{-1}\mathbf{Y}. \tag{2.6}$$

Here, we note that the graph Laplacian, $\mathbf{L}$, is a positive semi-definite matrix— all eigenvalues are non-negative, therefore, the largest eigenvalue of matrix $\mathbf{S} = \mathbf{I} - \mathbf{L}$ is 1. Coupling with the fact that ξ is in range $[0,1]$, it is shown that the following geometric series expansion converges to the middle term in (2.6):

$$\lim_{k\to\infty} \sum_{t=0}^{k-1} (\xi \mathbf{S})^t = (\mathbf{I} - \xi \mathbf{S})^{-1} \tag{2.7}$$

Then, it is possible to propose a $(k+1)$-th order approximation of the solution in 2.6 as follows (see Appendix A for the intermediate steps):

$$\mathbf{X}^{(k+1)} = \xi \mathbf{S}\mathbf{X}^{(k)} + (1-\xi)\mathbf{Y}. \tag{2.8}$$

Due to the iterative nature of this approximation, it is exploited by many algorithms in the literature, which will be mentioned in the forthcoming sections.

The representation model with ℓ_2 sense smoothness prior

The convergence of the iterative formulation in (2.8) suggests the following factor analysis on the inferred representations in terms of the matrix $\mathbf{S}$, which encodes the relational structure:

$$\mathbf{x} = \mathbf{S}\mathbf{x} + \boldsymbol{\epsilon}, \tag{2.9}$$

where $\mathbf{x} \in \mathbb{R}^N$ is a column vector of the embedding matrix $\mathbf{X}$. The reader might recognize that such a model is also referred to as structural equation model (SEM) [49], which describes the representation of a particular node on the graph as a linear combination of the ones belonging to its neighbors. In case where the initial node features are given randomly from a normal distribution, *i.e.*, $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_N)$, we obtain a multi-variate Gaussian distribution for

the representations, $\mathbf{x} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$, with a covariance

$$\boldsymbol{\Sigma} = (\sigma^2 \mathbf{I}_N - \mathbf{S})^{-1}. \tag{2.10}$$

Thus, we note that the ℓ_2 sense smoothness prior imposed on the representation leads to such a multivariate Gaussian as the generative model for the representations.

2.1.2 Neighborhood Aggregation

As indicated in the previous section, it is possible to converge to the optimal solution of Problem 1 in an iterative manner [39]. With a closer look on the iteration stated in (2.8), it can be seen that it updates the node representations by realizing an aggregation on the neighborhood structure, which is achieved by the first term on the right hand side. Here, the current state of the node representations $\mathbf{X}_k$ are multiplied by the matrix $\mathbf{S}$, which computes a linear combination of the representations of the local neighbors. This is because a row of $\mathbf{S}$ consists of zeros except at the indices corresponding to the first order (1-hop) neighbors of the node associated with that row, *i.e.*, $[\mathbf{S}]_{ij} = 0 \quad \forall\, j \neq i,\ (i,j) \notin \mathcal{E}$.

It is shown in Eqn. (2.9) that the node representations that are inferred by graph regularization with ℓ_2 sense smoothness prior fits a factor analysis model expressed by the graph structure globally. We can then write a local factor analysis model depending on partial correlation between two neighboring nodes as follows:

$$x_i = x_j + \epsilon, \tag{2.11}$$

where $(i,j) \in \mathcal{E}$ and $\epsilon \sim \mathcal{N}(0, \sigma_{ij}^2 \mathbf{I}_d)$, for $x_i \in \mathbb{R}^d$. The variance of the residual error, σ_{ij}^2, relates to the partial correlation between the neighbors, which is supposed to be given by the graphical model and d is the dimension of the vector representation of a node. The local model can be used to get an approximation of the node's representation in terms of its local neighborhood, which can be achieved by maximizing the expectation of the embedding at node-i given that of its 1-hop neighbors.

Problem 2: Bayesian estimation of the node representation by the local neighborhood

$$\underset{x_i}{\text{argmax}} \quad \text{p}(x_i | \{x_j : (i,j) \in \mathcal{E}\}) \tag{2.12}$$

Applying Bayes' rule, we obtain

$$\underset{x_i}{\text{argmax}} \quad \frac{\text{p}(\{x_j : (i,j) \in \mathcal{E}\} | x_i)\text{p}(x_i)}{\text{p}(\{x_j : (i,j) \in \mathcal{E}\})}. \tag{2.13}$$

Here, we make several assumptions in order to derive a first order approximation of the node's representation. First, we assume that the prior distribution on the node representations, $\text{p}(x_i)$

for $i \in \mathcal{V}$, is uniform. Second, we only consider the partial correlations between the central node—whose representation is to be estimated—and its 1-hop neighbors while we neglect any partial correlation among the neighborhood set—conditionally independence assumption. Accordingly, we reformulate the problem as

$$\underset{x_i}{\text{argmax}} \quad \prod_{(i,j)\in\mathcal{E}} \mathrm{p}(x_j|x_i), \tag{2.14}$$

which can also be stated by minimizing the negative log-likelihood as follows:

$$\underset{x_i}{\text{argmin}} \quad -\sum_{(i,j)\in\mathcal{E}} \log(\mathrm{p}(x_j|x_i)). \tag{2.15}$$

Using the local factor analysis model (2.11), we rewrite the problem (see Appendix A for the intermediate steps):

$$\underset{x_i}{\text{argmin}} \quad \sum_{(i,j)\in\mathcal{E}} \frac{\|x_j - x_i\|_2^2}{\sigma_{ij}^2}. \tag{2.16}$$

We note that the first order Bayesian estimate boils down to minimizing the Euclidean distance of the node's embedding to that of the neighboring nodes, *i.e.*, suggesting a least squares problem. This actually suits aforementioned ℓ_2 sense smoothness prior, aiming at minimizing ℓ_2 norm distance between connected node representations. Then, a first order Bayesian estimate is simply found by setting the gradient of the objective to zero:

$$\hat{x}_i = \frac{\sum_{(i,j)\in\mathcal{E}} \omega_{ij} x_j}{\sum_{(i,j)\in\mathcal{E}} \omega_{ij}}, \tag{2.17}$$

where $\omega_{ij} = 1/\sigma_{ij}^2$. We note that such a linear combination of neighbors is obtained through a first-order analysis of a node's representation only in the conditions considered above, which we referred to as conditional independence assumption. By this means, we account for the often used neighborhood aggregation operation with a Bayesian interpretation.

Following this analysis, we can finally write a local factor analysis model of a node's representation in terms of all neighbors, which reveals the uncertainty relating to the first order estimation of the node's representation.

$$x_i = \frac{\sum_{(i,j)\in\mathcal{E}} \omega_{ij} x_j}{\sum_{(i,j)\in\mathcal{E}} \omega_{ij}} + \epsilon \tag{2.18}$$

where the error variance of the aggregated error is calculated as

$$\frac{\sum_{(i,j)\in\mathcal{E}} \omega_{ij}^2 \sigma_{ij}^2}{\left(\sum_{(i,j)\in\mathcal{E}} \omega_{ij}\right)^2} = \frac{1}{\sum_{(i,j)\in\mathcal{E}} \omega_{ij}}. \tag{2.19}$$

Therefore, the error relating to the first order estimation is expressed as $\epsilon \sim \mathcal{N}(0, \frac{1}{\sum_{(i,j)\in\mathcal{E}} \omega_{ij}} \mathbf{I}_d)$.

Here, we draw attention to the fact that the obtained estimate is a linear combination of the neighbors' representation vectors. Therefore, the first order Bayesian estimate confirms the neighborhood aggregation operation accomplished in one step of the iterative formulation in (2.8). This implies that propagating the estimated representations across the whole graph in an iterative manner, it is possible to converge to the optimal solution of the graph regularization problem. In the next parts, we will mention some fundamental approaches adopting such a propagation technique. Then, in the next chapter, we will see how we adapt the first order Bayesian estimate in a multi-relational neighborhood to propose a propagation algorithm on a multi-relational graph.

Iterative Graph Algorithms

The iterative formulation of the graph regularization solution (2.8) is inherently used by seminal graph algorithms such as *PageRank* [50] and *Label Propagation* [51, 48] and *Random Walks* [52, 53].

Random Walks

The random walk-based approaches employ transition probabilities on the graph edges in order to compute the neighborhood aggregation and estimate the new node representations. Let us consider a weighted graph where a weight is assigned to each edge on the graph, indicating a measure of similarity between the connecting nodes. Here, the probability of transition from node-j to node-i is denoted by p_{ij} and it can be written in terms of the edge weights as follows:

$$p_{ij} = \frac{\omega_{ij}}{\sum_{(i,k)\in\mathcal{E}} \omega_{ik}}, \tag{2.20}$$

where ω_{ij} is the weight of the edge between node i and j. It is worth to notice that the first order Bayesian estimate in (2.17) can be framed as a linear combination of the neighboring node representations using these transition probabilities as

$$\hat{x}_i = \sum_{(i,j)\in\mathcal{E}} p_{ij} x_j, \tag{2.21}$$

An iteration of the random walk process can also be shown in matrix format as $\mathbf{x}^{(k+1)} = \mathbf{P}\mathbf{x}^{(k)}$ where $\mathbf{P}$ is the row-stochastic transition matrix, *i.e.*, elements in a row are summed up to unity, $\sum_{(i,j)\in\mathcal{E}} p_{ij} = 1$. This can simply be computed by $\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$ where $\mathbf{W}$ is the weight matrix of the graph. In order to guarantee the convergence, a *lazy random walk* process can be adopted by modifying the transition matrix as

$$\mathbf{T} = \xi\mathbf{P} + (1-\xi)\mathbf{I}_N \tag{2.22}$$

with $\xi \in (0,1)$. We also note the similarity of such an update step to the iterative formulation of the graph regularization solution (2.8).

PageRank

The PageRank algorithm is akin to random walks since it also employs probabilities assigned to the links between pages and estimates the likelihood of jumping from one page to another. In this sense, the transition matrix in (2.22) can be recruited to update the page-rank values π as

$$\pi = \mathbf{T}\pi. \tag{2.23}$$

Similarly, p_{ij} holds the probability of jumping from page-j to page-i and the hyperparameter ξ in (2.22) is known as the damping factor, accordingly, $(1-\xi)$ can be considered as the rate of favoring the current position.

Label Propagation

The iterative neighborhood aggregation is also practiced by the line of works in graph-based semi-supervised learning, which performs inference with partially labeled data. Label propagation achieve this by transmitting the label information from the nodes whose label is known towards the ones whose label is unknown, across the relational structure. For this purpose, the study in [51] proposes a simple iterative algorithm using the probabilistic transition matrix $\mathbf{P}$ that was introduced previously. The propagation step of the algorithm can be expressed as

$$\mathbf{Y} = \mathbf{PY}, \tag{2.24}$$

where matrix $\mathbf{Y}$ stores the label information in terms of one-hot coding. This step is followed by the normalization of the updated label matrix and then clamping the initially labeled data. The last step leaves initially known labels as unchanged and impose them to rejoin at every iteration as they are. Therefore, this operation can easily be expressed in the iterative format of graph regularization, where matrix $\mathbf{S}$ is then replace by the transition matrix.

Neural Network Schemes on Graphs

The representations suggested by the graph regularization scheme can be obtained through a series of linear operations on the neighborhood structure, thus it hinders capturing complex, non-linear features. Similar to the iterative propagation algorithms, the neural network schemes defined on graphs propagate node representations along the edges of the graph. The representations are re-computed at each layer of the neural net and transferred to the next layer. The fundamental difference between these two lines of works lies in the parameterization of the neighborhood aggregations and the inclusion of nonlinear activation functions between the layers. This actually promotes a deeper learning architecture of the neural net which raises the expressive power of the representations.

It is important to note that the neural nets on graphs ultimately accomplish an information exchange over graph's nodes through its edges, thus, they recruit the given relational structure of the data as the computational graph. In this regard, the authors in [54] emphasize that such

a message-passing operation is encountered both in label propagation algorithms and neural nets on graphs and this can be viewed as feature/label smoothing. This approach smooths the features, which leads to smoothing the predictions by spreading out the error [55]. Hence, the intuition behind these two lines of works is error smoothing, which relies on the assumption that the errors on connected nodes are positively correlated. In fact, this resonates with the smoothness assumption that we initially stated: "Neighboring nodes should have similar representations.".

In this section, we introduce notable neural network schemes for graph representation learning. Next, we briefly mention the learning schemes developed for multi-relational or heterogeneous graphs, which hints the progress of representation learning on more complex and relational data domains.

Graph Neural Network Model. A neural network scheme on graphs is first introduced in [56, 57] under the name of Graph Neural Network (GNN), which was referred to prominently in the following decade. The GNN model is designed as a recurrent neural network composed of repeated application of propagation functions. Therefore, it can be framed in two steps [58]: propagation and output. The operations in these steps incorporate any existing node and edge labels, including the edge directions on the graph. In this sense, it can actually be applied for learning node embedding vectors on heterogeneous networks consisting of multiple types of nodes and edges. The propagation step in GNN model is formulated as follows.

$$x_i^{(k+1)} = f\Big(\mathtt{l}_i, \{(x_j^{(k)}, \mathtt{l}_{(i,j)}, \mathtt{l}_j) \quad \forall (i,j) \in \mathcal{E}\}\Big), \tag{2.25}$$

where $x_i^{(k)}$ is the node representation of node-i at layer-k. Also, $\mathtt{l}_i$ stands for the label of node-i, and $\mathtt{l}_{(i,j)}$ stands for the label and direction information for the edge (i,j). Then, f is denoted as the local transition function which calculates node's representation from its neighbors, and it is parameterized with respect to node's label, neighbors' labels and the label and direction information of the edges connecting to the neighbors. Finally in the output step, the output o_i is produced from the final node representations, $x_i^{(K)}$—for a K layer GNN:

$$o_i = g(\mathtt{l}_i, x_i^{(K)}). \tag{2.26}$$

The output function g is parameterized with respect to node's label. We note that it is possible to express the propagation and output function globally on the graph and modify the output step with respect to the downstream task such as node-level or graph-level regression, classification etc.

Graph Convolution Framework. Graph convolution framework adapts the convolution kernel defined on regular grid structure to the irregular neighborhood structure of graphs. This again boils down to aggregation of the node representations within a certain neighborhood and updating them through the layers of neural scheme called Graph Convolution Network

(GCN). At k^{th} layer of a GCN, the forward model is given as follows:

$$\mathbf{X}^{(k+1)} = f^{(k)}(\mathbf{X}^{(k)}, \mathbf{A}) = \sigma(g(\mathbf{A})\mathbf{X}^{(k)}\boldsymbol{\Theta}^{(k)}) \tag{2.27}$$

where $\mathbf{X}^{(k)}$ is the matrix storing node embeddings in its rows and initialized with some input node features. Here, $\boldsymbol{\Theta}^{(k)}$ applies a linear feature transformation on the representation matrix and it constitutes the learnable parameters at layer-k of the neural net. Then, $\sigma(\cdot)$ is a non-linear function such as ReLU (rectified linear unit), and $g(\cdot)$ is a graph kernel which yields the convolution operator for the given adjacency matrix, $\mathbf{A}$, of the input graph structure. The convolution operation often appears as the aggregation of the representations accommodated at direct neighbors [59]. Nonetheless, it is possible to obtain a linear combination of higher order neighbors, especially when the convolution kernel is defined on the spectral domain of the graph [60, 3, 61]. In this case, $g(\mathbf{A})$ performs an aggregation on the multi-hop neighborhood by approximating a filtering function expressed on the eigenbasis of the graph Laplacian.

It is worth to notice that the convolution operation on graphs treats every neighbor equally in the aggregation, which is inherently isotropic. In fact, it is not straightforward to obtain anisotropic operations on graphs unlike on grids. This is because graphs are irregular structures by nature, and it is not clear how to define a local notion of orientation. Recently advancing anisotropic models are favorable in terms of the expressivity of the representations [62]. For instance, graph attention network (GAT) [63] permits treating the neighbors differently in the aggregation by learning attention weights for them. In this sense, augmenting the computational graph by exploiting the available heterogeneous, multi-relational knowledge provided by the data domain hints a promising direction for boosting the representational power of the learning scheme.

Message-Passing Framework. The essential steps accomplished by the previously introduced iterative algorithms and neural nets on graphs can also be reframed as message passing between the neighbors, aggregation of the collected messages and re-computation of the node representations. To this end, Gilmer et al. [4] introduce a general framework for supervised learning on graphs that is called as Message Passing Neural Network (MPNN), where a message-passing layer is expressed by the following formulation:

$$m_i^{(k)} = g\Big(\{f^{(k)}(x_i^{(k)}, x_j^{(k)}, \mathtt{l}_{ij}) \quad \forall (i,j) \in \mathcal{E}\}\Big) \tag{2.28}$$

$$x_i^{(k+1)} = \sigma^{(k)}(x_i^{(k)}, m_i^{(k)}), \tag{2.29}$$

where $f^{(k)}(\cdot)$ is the message function at layer-k and the function $g(\cdot)$ aggregates the messages collected from the neighbors. Then, $\sigma^{(k)}(\cdot)$ is the update function at layer-k, which combines the current node representation with the aggregated messages, $m_i^{(k)}$, and re-compute the node representation for the next layer. We note that the function f determines the message by taking both source node's (node-j) and target node's (node-i) representation into account. In addition, it allows to input available edge features $\mathtt{l}_{ij}$ that can be represented as edge embedding vector and also be learned. In this regard, the message passing framework provides

a flexibility in adapting to complex and heterogeneous graphs consisting of different types of nodes and edges. Originally, function g is introduced simply as a sum aggregation, yet it is possible to generalize it as a parametric function as well.

Later, many GNN works benefited from the message-passing framework in order to frame their forward learning scheme. Omitting the customized message construction proposed by MPNN, the forward model can easily be reduced to aggregate and combine steps [64, 65]:

$$x_i^{(k+1)} = \text{COMBINE}\Big(x_i^{(k)}, \text{AGGREGATE}\big(\{x_j^{(k)} \quad \forall (i,j) \in \mathcal{E}\}\big)\Big). \tag{2.30}$$

AGGREGATE and COMBINE can be some parametric functions special to each layer and including non-linearities.

Learning on Heterogeneous and Multi-relational Graphs. Recent years have witnessed a rise in real-world data that is captured with rich structural information, which can be better depicted by heterogeneous or multi-relational graphs. In contrast, the research on graph representation learning was persistent on the problems arising in simple, homogeneous graphs in the past decade. Nonetheless, there have been several extensions of the existing GCN models in multi-relational settings such as R-GCN[66, 67] and R-GAT [68, 69]—the prefix R stands for relational. They typically learn relation-specific parameters for feature transformation or aggregation. For comparison, we summarize the aggregation functions employed by GCN, GAT models and their relational versions as

$$\text{GCN:} \quad \sigma\Big(\sum_{(i,j)\in\mathcal{E}} \boldsymbol{\Theta}^\top x_j\Big), \qquad \text{R-GCN:} \quad \sigma\Big(\sum_{(i,j)\in\mathcal{E}} \boldsymbol{\Theta}_{\mathtt{r}(i,j)}^\top x_j\Big), \tag{2.31}$$

$$\text{GAT:} \quad \sigma\Big(\sum_{(i,j)\in\mathcal{E}} \alpha(i,j)\,\boldsymbol{\Theta}^\top x_j\Big), \qquad \text{R-GAT:} \quad \sigma\Big(\sum_{(i,j)\in\mathcal{E}} \alpha_{\mathtt{r}(i,j)}(i,j)\,\boldsymbol{\Theta}_{\mathtt{r}(i,j)}^\top x_j\Big), \tag{2.32}$$

where aggregation applied on the neighborhood of node-i. $\mathtt{r}(i,j)$ indicates the relation type between node-i and j. Accordingly, in the relational versions, the feature transformation matrix is diversified with respect to the relation type as $\boldsymbol{\Theta}_{\mathtt{r}(i,j)}$. Similarly, the attention weights, $\alpha(i,j)$, used in GAT varies with respect to the relation type in R-GAT.

For data relying on heterogeneous graphs, besides structural information, the complexity of the feature information requires special attention to handle. To illustrate, different types of nodes may possess different types of properties that can be expressed in different feature spaces. Their incorporation in the learning scheme simultaneously with the graph still remains an open challenge. In [70], the authors addressed this issue by proposing type-specific transformation matrix for mapping different types of node features into the same feature space. Then, they adopt a hierarchical and relational attention mechanism to aggregate the neighbors emerging from different types of meta-paths on a heterogeneous graph.

At this point, it is worth to note that directly augmenting the number of learning parameters with respect to the volume of multi-relational information could be problematic. This is because the number of relation types usually increases with not enough number occurrences

on the graph, which may cause over-parameterization and instable training process. Similarly, the diversity of the message passing paths expands too fast with the increasing number of nodes and edge types in a heterogeneous graph. To overcome this, *heterogeneous mutual attention* is proposed in [71] where the attention weight of a meta-relation (a message path) is decomposed with respect to the source node type, target node type and the edge type between them. Such a parameter sharing in the attention mechanism is reported to provide a generalization for learning over heterogeneous graphs.

2.2 Inverse Problem: Inference of the Underlying Graph Structure

In the previous section, we discussed the problem of inferring representations for a given relational structure. This section focuses on an inverse problem that targets inferring the latent relational structure underlying the data. Most of the studies on relational representation learning assume that the intrinsic structure of the data is readily available. However, the structure of the data can also be implicit, in which case it is required to be discovered from the observations. This is an important step for further data analysis and processing tasks such as capturing similarities and interactions within data and then semantic interpretation.

The relational inductive bias can be rephrased for the graph learning problem as "Nodes constituting similar observations should be neighbors.". Again, the main statistical prior comes from the smoothness assumption, which prescribes to search for a dependency structure minimizing the distance over the connected nodes. However, estimation of such a graph structure in high dimensional settings, where the number of nodes is higher than the number of observations, is ill-posed. In this case, further structural priors may apply, such as the sparsity of the graph to be learned. Then, the structure inference problem can be handled within the interplay of the sparsity and the smoothness.

Early works inferring the structure of the data indeed concentrated on the sparsity of partial correlations within data and imposed the relevant structural priors on the inverse covariance matrix of the data. This is because the zero entries in the inverse covariance matrix signify the conditional independency between variables and thus reveal the relational structure. The latter graph learning approaches exploited the notion of smoothness by defining smooth signals via a Fourier analysis on graphs. This approach aims at choosing the structure maximizing the smoothness of the observations or, to put it more mathematically, minimizing the Dirichlet energy (see the graph regularization loss that we obtain for ℓ_2 sense smoothness prior in Equation (2.4)).

2.2.1 Inverse Covariance Estimation

Dempster [72] introduced one of the earliest works on the selection of covariance for i.i.d. observations generated by a multivariate Gaussian distribution:

$$\mathbf{X}^\top = (x_1, \ldots, x_N) \sim \mathcal{N}(0, \boldsymbol{\Sigma}) \tag{2.33}$$

and proposed the idea of pruning the inverse covariance matrix in the quest for a sparse dependency structure.

The structure is usually characterized by a graphical model $\mathcal{G}(\mathcal{V}, \mathcal{E})$ where each variable is represented by a node in $\mathcal{V}$ and the edges between the nodes, $\mathcal{E}$, specify the dependency between variables. The model satisfies the Markov property: conditional independency between two variables given all the rest, $x_i \perp x_j \quad | \quad \mathbf{X}^\top / \{x_i, x_j\}$, is indicated by the lack of an edge between them, $(i, j) \notin \mathcal{E}$. In other words, their partial correlation is zero after removing the effect of all other variables. The inverse covariance matrix, $\boldsymbol{\Sigma}^{-1} = \boldsymbol{\Theta}$—also called the precision matrix—of the data is decisive for identifying the graph structure since it measures such partial correlations within data. Namely, if $\boldsymbol{\Theta}_{ij} = 0$ then $(i, j) \notin \mathcal{E}$.

With this in mind, Meinhausen & Bühlmann [73] designate each variable as a linear combination of its neighbors and present neighborhood selection as a subproblem of covariance selection. Accordingly, they estimate the conditional dependency for each node separately by solving a regression problem

$$\min_{\theta_i} \|x_i - \mathbf{X}^\top \theta_i\|_2^2 + \lambda \|\theta_i\|_1, \tag{2.34}$$

where the second term is a Lasso regularizer on the neighbor coefficients, θ_i for node-i. This is due to one of their main assumptions: the graph structure is sparse, which enforces a restriction on the neighborhood size.

An important representative of the sparse inverse covariance estimation methods is graphical Lasso introduced by Friedman et al. [74]. They infer the precision matrix all at once via maximizing its likelihood on the Gaussian graphical model (2.33):

$$\mathrm{p}(\mathbf{X}|\boldsymbol{\Theta}) \propto \det(\boldsymbol{\Theta})^{\frac{1}{2}} \exp(-\frac{1}{2}\mathrm{tr}(\hat{\boldsymbol{\Sigma}}\boldsymbol{\Theta})), \tag{2.35}$$

where $\hat{\boldsymbol{\Sigma}}$ is the empirical covariance. Then, they achieve the estimation by including a Lasso regularizer on the precision matrix:

$$\max_{\boldsymbol{\Theta}} \quad \mathrm{logdet}(\boldsymbol{\Theta}) - \mathrm{tr}(\hat{\boldsymbol{\Sigma}}\boldsymbol{\Theta}) - \lambda \|\boldsymbol{\Theta}\|_1. \tag{2.36}$$

Such a coupled optimization on the variables undertake a more stable solution compared to the neighborhood selection problem, although it involves a log determinant term, which is computationally demanding.

2.2.2 Estimation of Graph Laplacian

At this point, we refer back to Section 2.1.1 where we obtain a representation model with ℓ_2 sense smoothness prior by following the graph regularization approach. We remind that we end up with a Gaussian generative model which relates the precision matrix to the neighborhood structure. In particular, Equation (2.10) reveals the connection between the precision matrix and the graph Laplacian. As a matter of fact, the graph Laplacian is a singular matrix which identifies the graph structure uniquely. With this in mind, Lake & Tenenbaum [75] introduce an interpolation between precision matrix of the data and the graph Laplacian, then propose the following optimization problem that is akin to graphical Lasso:

$$\max_{\boldsymbol{\Theta},\sigma^2} \quad \operatorname{logdet}(\boldsymbol{\Theta}) - \operatorname{tr}(\boldsymbol{\Theta}\mathbf{X}\mathbf{X}^\top) - \lambda\|\mathbf{W}\|_1 \tag{2.37}$$

$$\text{subject to} \quad \boldsymbol{\Theta} = \mathbf{D} - \mathbf{W} + \frac{1}{\sigma^2}\mathbf{I}, \tag{2.38}$$

where the precision matrix $\boldsymbol{\Theta}$ can be seen as a regularized Laplacian. This approach is noteworthy in terms of discerning feature smoothness as an optimization over the precision matrix decomposing the sample covariance, which appears as the trace term in the objective (2.37). We remark that likewise, this term is often encountered as quadratic Laplacian form (2.4) in graph regularization problems.

Smooth Signal Representation Model

As signals can be generalized with statistical models regarding their frequency components, it is possible to define signals on graphs and draw an analogy from signal processing on regular domains to the signal processing on graphs [28]. Indeed, a *graph signal* can be denoted by a vector that collects the nodal observations across the entire graph, *i.e.*, $\mathbf{x} \in \mathbb{R}^N$ for $|\mathcal{V}| = N$. Then, eigenvectors of the Laplacian matrix of a graph provide a basis to express any signal defined on that graph,

$$\mathbf{L} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\top.$$

Each column of the eigenbasis $\mathbf{Q}$ can be interpreted as a component of the graph signal, ordered from low frequency to high, which is associated with the eigenvalues—diagonal elements of $\boldsymbol{\Lambda}$—sorted in an increasing order. An important property of natural signals represented on graphs is the fact that they change smoothly on their graph structure. This inherently relates to a signal decomposition where the low-frequency graph components encode slow variations across the neighborhood structure of the graph, whereas higher ones hold more complex patterns. Leveraging such a graph Fourier analysis, Dong et al. [76] propose a factor analysis model for graph signals defined in terms of frequency coefficients $\mathbf{h}$:

$$\mathbf{x} = \mathbf{Q}\mathbf{h} + \boldsymbol{\epsilon} \tag{2.39}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$. Then the following statistical model is given for the smooth signals on the graph[I]:

$$\mathbf{h} \sim \mathcal{N}(0, \boldsymbol{\Lambda}^{\dagger}) \tag{2.40}$$

Coupled with the factor analysis in (2.39), it follows the same Gaussian graphical model obtained for the representations with ℓ_2 sense smoothness prior in Section 2.1.1:

$$\mathbf{x} \sim \mathcal{N}(0, \mathbf{L}^{\dagger} + \sigma^2 \mathbf{I}).$$

Now, one can exploit such a smooth signal representation model as a prior on the observations while maximizing the a posteriori estimate of the graph Laplacian. Accordingly, this optimization problem is formulated as follows:

$$\min_{\mathbf{L},\mathbf{Y}} \quad \|\mathbf{X}-\mathbf{Y}\|_F^2 + \alpha \mathrm{tr}(\mathbf{Y}^{\top}\mathbf{L}\mathbf{Y}) + \beta \|\mathbf{L}\|_F^2 \tag{2.41}$$

$$\text{subject to} \quad \mathrm{tr}(\mathbf{L}) = N, \mathbf{L} \in \mathcal{L}, \tag{2.42}$$

where $\mathcal{L}$ is set of valid graph Laplacians, namely that are symmetric and satisfy zero row-sum. The trace constraint can be considered as a budget on the volume of the graph which will be distributed as the weights on the graph edges. Since the trace term in the objective automatically impose the graph sparsity, a Frobenius norm on the Laplacian is used as a regularizer. Thus, the hyperparameters adjusts the sparsity of the solution. Then, $\mathbf{Y}$ is composed of a set of smooth signals on the graph and interpolates the observed signals in $\mathbf{X}$. The problem can be solved via quadratic programming within an alternating minimization scheme.

Moreover, relevant generative models emerged from a diffusion process are studied in [77, 78], where they recover a network topology from the eigenbasis of a graph shift operator such as the graph Laplacian. A more detailed categorization of the graph signal processing based approaches can be found in [79, 34].

[I] † stands for Moore-Penrose pseudo-inverse.

3 Multi-Relational Propagation for Node Regression

In the previous chapter, we mention graph regularization and neighborhood aggregation methods for node representation learning in simple, homogeneous graphs. Developing on that, this chapter provides a passage to node-level inference on multi-relational graphs. Here, we introduce a multi-relational representation learning model by focusing on the node regression task in transductive settings. In particular, we consider the following problem. Given the multi-relational structure of the data, we aim at completing the missing node features. To this end, we first provide a relational local generative model which leads to aggregation on a multi-relational and directed neighborhood. Next, building on top of that, we propose an iterative neighborhood aggregation method for node regression, which we call multi-relational propagation algorithm, MRP. In this regard, our method can be considered as a sophisticated version of the well-known label propagation algorithm [51] by enabling operation on a multi-relational and directed graph.

This chapter is organized based on the work titled “Propagation on Multi-relational Graphs for Node Regression" [80].

Comparison to the previous approaches. Node regression problem has been studied on simple and homogeneous graphs for signal inpainting on graphs [81, 82] and node representation learning [83, 84, 85, 86]. Also, we refer the reader to the previous chapter, Section 2.1, to revisit the representation learning methods working on simple graphs. In our review, we emphasize that the learning methods working on simple graphs mainly exploit ℓ_2 sense smoothness, which prescribes minimizing the Euclidean distance between features at the connected nodes. Despite its practicality, this approach suffers from several major limitations which might mislead regression on a multi-relational and directed graph. First, it treats all neighbors of a node equally during the inference about the node's state, although neighbors connected via different types of relations might play a different role in the inference task. For instance, Figure 3.1 illustrates multiple types of relationships that might arise between people. Here, each relation type presumably relies on different affinity rules or different levels of importance depending on the node regression task. It is also worth to notice that some relation types are inherently symmetric such as `sibling`, whereas some others are asymmetric such as `parent`.

This is also indicated by the direction of the graph edges, see Figure 3.1. Euclidean distance minimization broadly assumes the values at the neighboring nodes are as close as possible, which may not always be the case. Thus, the inference approach applied on simple graphs is insufficient for handling the asymmetry emerging from the directed relationships.

Thus, we depart from the straightforward ℓ_2 sense of smoothness and augment the inductive bias with different types of relationships within data. Accordingly, we present a novel local generative model for a multi-relational neighborhood which leads to a neighborhood aggregation operation depending on relational transformations and facilitates the iterative steps of the proposed propagation algorithm MRP.

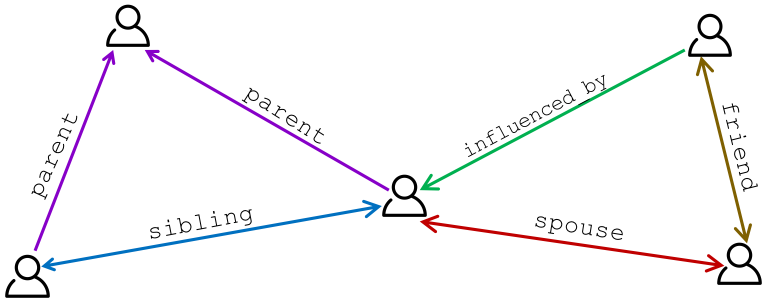


Figure 3.1 – A fragment of a multi-relational and directed social network

3.1 Multi-relational Model

We first introduce the settings and the notation that we study the node regression problem. We denote a multi-relational and directed graph as $\mathcal{G}(\mathcal{V},\mathcal{E},\mathcal{P})$, where $\mathcal{V}$ is the set of nodes, $\mathcal{P}$ is the set of relation types, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{P} \times \mathcal{V}$ is the set of multi-relational edges. The function $\mathtt{r}(i,j)$ returns the relation type $\mathtt{p} \in \mathcal{P}$ that is pointed from node j to node i. If such a relation exists between them, yet pointed from the node i to the node j, then the function returns the reverse as $\mathtt{p}^{-1}$.

Relational Local Generative Model. We recall that in Section 2.1.2, we revisit the local generative model on a simple, homogeneous neighborhood, adopting ℓ_2 sense smoothness prior (2.11). That model preserves the smoothness by inhibiting the change between the neighboring node representations. Similarly, we build our inductive bias by minimizing the change over neighboring nodes. However, in a multi-relational structure, it is required to diversify the local generative model by the set of relationships existing on the graph. To this end, we propose the following local generative model for the node given its multi-relational and directed neighbors:

$$x_i = \begin{cases} \eta_{\mathtt{p}} x_j + \tau_{\mathtt{p}} + \epsilon, & \forall \mathtt{r}(i,j) = \mathtt{p} \text{ where } \epsilon \sim \mathcal{N}(0, \sigma_{\mathtt{p}}^2) \\ \dfrac{x_j}{\eta_{\mathtt{p}}} - \dfrac{\tau_{\mathtt{p}}}{\eta_{\mathtt{p}}} + \epsilon, & \forall \mathtt{r}(i,j) = \mathtt{p}^{-1} \text{ where } \epsilon \sim \mathcal{N}(0, \dfrac{\sigma_{\mathtt{p}}^2}{\eta_{\mathtt{p}}^2}), \end{cases} \tag{3.1}$$

where $x_i \in \mathbb{R}$ denotes the value assigned to node-i. Equation (3.1) builds a linear relationship between the neighboring nodes by introducing relation-dependent scaling parameter η and a shift parameter τ. The latter case in (3.1) indicates the generative model yielded by the reverse relation, where the direction of the edge is reversed with respect to the former, thus, it is simply the reverse of the equation in the former case. We note that the proposed linear model conforms both symmetric and asymmetric relationships. This is because it can capture any bias over a certain relation through parameter τ or even any change in scale through parameter η. We note that the default set for these parameters are suggested as $\tau = 0, \eta = 1$, which boils down to the local generative model on simple graphs given in Eqn. (2.11).

3.1.1 First-order Relational Bayesian Estimate

At this point, we again refer to Chapter 2 where we derive the first order Bayesian estimate of a node representation (2.17) given its immediate neighbors on a simple graph. Likewise, using the proposed relational local generative model (3.1), it is possible to estimate the node value through its first-hop neighbors which are connected via multiple types relationships.

For this purpose, we consider the following settings. First, we assume uniform prior distribution on the node values. Second, we grant the first-hop connections of the central node—whose state is to be estimated, while we neglect any connection that might originate from the further neighborhood. In these settings, we cast the problem as maximizing the likelihood of node's immediate neighbors, which then can be written in product of likelihood of each neighbor, similar to the derivation in simple graphs (2.14).

We now integrate the proposed local generative model (3.1) in the estimation problem. To begin with, one can express the likelihood of a relational neighbor as follows:

$$\mathrm{p}(x_j|x_i) = \begin{cases} \sqrt{\frac{\omega_{\mathtt{p}}}{2\pi}} \exp\left(-\frac{\omega_{\mathtt{p}}}{2}\Big(x_i - \eta_{\mathtt{p}} x_j - \tau_{\mathtt{p}}\Big)^2 \right), & \forall \mathtt{r}(i,j) = \mathtt{p} \\ \sqrt{\frac{\omega_{\mathtt{p}}\eta_{\mathtt{p}}^2}{2\pi}} \exp\left(-\frac{\omega_{\mathtt{p}}\eta_{\mathtt{p}}^2}{2}\Big(x_i - \frac{x_j}{\eta_{\mathtt{p}}} + \frac{\tau_{\mathtt{p}}}{\eta_{\mathtt{p}}}\Big)^2 \right), & \forall \mathtt{r}(i,j) = \mathtt{p}^{-1}, \end{cases} \tag{3.2}$$

where we apply a change of parameter $\omega_p = 1/\sigma_{\mathtt{p}}^2$. Next, the estimation can be found by minimizing the negative log-likelihood as in (2.15). Once, the likelihoods (3.2) are substituted, we obtain the following objective.

Problem 1: Bayesian estimation of the node's state with relational local neighborhood

$$\underset{x_i}{\operatorname{argmin}} \quad \sum_{\mathtt{p}\in\mathcal{P}} \left(\sum_{\mathtt{r}(i,j)=\mathtt{p}} \frac{\omega_{\mathtt{p}}}{2}\Big(x_i - \eta_{\mathtt{p}} x_j - \tau_{\mathtt{p}}\Big)^2 + \sum_{\mathtt{r}(i,j)=\mathtt{p}^{-1}} \frac{\omega_{\mathtt{p}}\eta_{\mathtt{p}}^2}{2}\Big(x_i - \frac{x_j}{\eta_{\mathtt{p}}} + \frac{\tau_{\mathtt{p}}}{\eta_{\mathtt{p}}}\Big)^2 \right). \tag{3.3}$$

For an arbitrary node $i \in \mathcal{V}$, we denote the loss to be minimized as $\mathcal{L}_i$. Such a loss leads to a least squares problem whose solution satisfies $\dfrac{\partial \mathcal{L}_i}{\partial x_i}(\hat{x}_i) = 0$, the gradient and the intermediate

step to the solution are given in Appendix B. Accordingly, the estimate can be found as

$$\hat{x}_i = \frac{\sum_{\mathrm{p}\in\mathcal{P}} \left(\sum_{\mathtt{r}(i,j)=\mathrm{p}} \omega_{\mathrm{p}} \left(\eta_{\mathrm{p}} x_j + \tau_{\mathrm{p}} \right) + \sum_{\mathtt{r}(i,j)=\mathrm{p}^{-1}} \omega_{\mathrm{p}} \eta_{\mathrm{p}} \left(x_j - \tau_{\mathrm{p}} \right) \right)}{\sum_{\mathrm{p}\in\mathcal{P}} \left(\sum_{\mathtt{r}(i,j)=\mathrm{p}} \omega_{\mathrm{p}} + \sum_{\mathtt{r}(i,j)=\mathrm{p}^{-1}} \omega_{\mathrm{p}} \eta_{\mathrm{p}}^2 \right)}. \tag{3.4}$$

3.1.2 Estimation of Relational Parameters

The parameters of the local generative model associated with relation type $\mathrm{p} \in \mathcal{P}$ are introduced as $\{\tau_{\mathrm{p}}, \eta_{\mathrm{p}} \omega_{\mathrm{p}}\}$. These parameters can be estimated over the set of node pairs connected to each other by relation p, *i.e.*, $\{(x_i, x_j)\ \forall i, j \in \mathcal{V} \mid \mathtt{r}(i,j) = \mathrm{p}\}$. For this purpose, we carry out the maximum likelihood estimation over the parameters:

$$\underset{\tau_{\mathrm{p}}, \eta_{\mathrm{p}} \omega_{\mathrm{p}}}{\operatorname{argmax}} \quad \mathrm{p}\Big(\{(x_i, x_j)\ \forall i, j \in \mathcal{V} \mid \mathtt{r}(i,j) = \mathrm{p}\} \,\Big|\, \tau_{\mathrm{p}}, \eta_{\mathrm{p}} \omega_{\mathrm{p}} \Big) \tag{3.5}$$

Then, we conduct an approximation over the node pairs that are connected by a given relation type while neglecting any conditional dependency that might exist among these node pairs [I]. Hence, we can write the likelihood on each node pair in a product as follows:

$$\underset{\tau_{\mathrm{p}}, \eta_{\mathrm{p}} \omega_{\mathrm{p}}}{\operatorname{argmax}} \quad \prod_{\mathtt{r}(i,j)=\mathrm{p}} \mathrm{p}\Big((x_i, x_j) \,\Big|\, \tau_{\mathrm{p}}, \eta_{\mathrm{p}} \omega_{\mathrm{p}} \Big) \tag{3.6}$$

Then, the likelihood of a pair of values (x_i, x_j) belonging to the nodes connected by relation type p given the parameters of the associated generative model (3.1) can be expressed as follows:

$$\mathrm{p}\Big((x_i, x_j) \mid \mathtt{r}(i,j) = \mathrm{p} \,\Big|\, \tau_{\mathrm{p}}, \eta_{\mathrm{p}} \omega_{\mathrm{p}} \Big) = \sqrt{\frac{\omega_{\mathrm{p}}}{2\pi}} \exp\left(-\frac{\omega_{\mathrm{p}}}{2} \Big(x_i - \eta_{\mathrm{p}} x_j - \tau_{\mathrm{p}} \Big)^2 \right). \tag{3.7}$$

Accordingly, we proceed with the minimization of negative log-likelihood to solve the problem in (3.6). The reader might recognize that the solution of this problem is equivalent to the parameters of a linear regression model [87]. This is simply because we introduce linear generative models (3.1) for the relationships existing on the graph. Therefore, the parameters of the generative model can be found as follows:

$$\eta_{\mathrm{p}} = \frac{\sum_{\mathtt{r}(i,j)=\mathrm{p}} (x_i - \mu)(x_j - \mu)}{\sum_{\mathtt{r}(i,j)=\mathrm{p}} (x_j - \mu)^2}, \tag{3.8}$$

[I] A first-order approximation is conducted where each node pair connected via a certain relation type is considered as an independent observation in the parameter estimation of that relation. Although these node pairs might appear in the same neighborhood, any correlation between them is neglected. Also the parameter set of each relation type is estimated separately from the other relationships.

where $\mu = \text{mean}(\mathbf{x})$ is the mean of the node values. Then,

$$\tau_{\mathrm{p}} = \text{mean}\Big(\{(x_i - \eta_{\mathrm{p}} x_j)\ \forall i,j \in \mathcal{V} \mid \mathtt{r}(i,j) = \mathrm{p}\}\Big), \tag{3.9}$$

$$\omega_{\mathrm{p}} = 1/\text{mean}\Big(\{(x_i - \eta_{\mathrm{p}} x_j - \tau_{\mathrm{p}})^2\ \forall i,j \in \mathcal{V} \mid \mathtt{r}(i,j) = \mathrm{p}\}\Big). \tag{3.10}$$

Intermediate steps in the derivation of the parameters can be found in the Appendix B.

Local Generative Model and Local Operation. We now recap the proposed multi-relational inference approach in comparison to inference on simple, homogeneous graphs analyzed in Chapter 2, Section 2.1.2. For this purpose, we summarize the local generative model, the loss associated with the estimation and the corresponding first order estimate for both cases in Table 3.1.

Table 3.1 – Local Generative Model and Operation in Simple and Multi-relational Graphs

	Local Generative Model	**Loss**	**Local Operation**
Simple Weighted Graph	$x_i = x_j + \epsilon$ $\forall (i,j) \in \mathcal{E}$ $\epsilon \sim \mathcal{N}(0, 1/\omega_{ij})$	$\sum_{(i,j)\in\mathcal{E}} \omega_{ij}(x_i - x_j)^2$	$\frac{\sum_{(i,j)\in\mathcal{E}} \omega_{ij} f(x_j)}{\sum_{(i,j)\in\mathcal{E}} \omega_{ij}}$
Multi-relational Directed Graph	$x_i = \eta_{\mathrm{p}} x_j + \tau_{\mathrm{p}} + \epsilon$ $\forall \mathtt{r}(i,j) = \mathrm{p}$ $\epsilon \sim \mathcal{N}(0, 1/\omega_{\mathrm{p}})$	$\sum_{\mathrm{p}\in\mathcal{P}\cup\mathcal{P}^{-1}} \sum_{\mathtt{r}(i,j)=\mathrm{p}} \omega_{\mathrm{p}}(x_i - \eta_{\mathrm{p}} x_j - \tau_{\mathrm{p}})^2$	$\frac{\sum_{\mathrm{p}\in\mathcal{P}\cup\mathcal{P}^{-1}} \sum_{\mathtt{r}(i,j)=\mathrm{p}} \omega_{\mathrm{p}} f_{\mathrm{p}}(x_j)}{\sum_{\mathrm{p}\in\mathcal{P}\cup\mathcal{P}^{-1}} \sum_{\mathtt{r}(i,j)=\mathrm{p}} \omega_{\mathrm{p}}}$

The first row in Table 3.1 summarizes the inference on a simple, weighted graph, where the local generative model built with ℓ_2 sense smoothness prior. This leads to minimizing the Euclidean distance between the connected node pairs—the associated loss. The second row states the proposed relational local generative model, which leads to minimizing not the Euclidean distance directly but the distance calculated upon a transformation applied on the neighbor.

In the table, we frame the first order relational Bayesian estimate, which is expressed in (3.4), in a neighborhood aggregation. Unlike in the simple case, it is not a straightforward weighted average of the neighbors. However, the neighbors are subject to a transformation with respect to the type and the direction of their relation to the central node. The relational transformation is controlled by the parameters η and τ. For this reason, in Table 3.1 we use the following functions as shortcuts for the transformations applied on the neighbors in simple and multi-relational case:

$$\begin{aligned} f(x) &= x, && \text{in simple case, no actual transformation applied,} \\ f_{\mathrm{p}}(x) &= \eta_{\mathrm{p}} x + \tau_{\mathrm{p}}, && \text{in relational case for type p.} \end{aligned}$$

In addition, $\mathcal{P}^{-1} = \{\mathtt{p}^{-1}, \forall \mathtt{p} \in \mathcal{P}\}$ denotes the set relation types where the edge direction is reversed. For the reversed relationships, the set of parameters can be simply set as follows:

$$\eta_{\mathtt{p}^{-1}} = \frac{1}{\eta_{\mathtt{p}}}, \qquad \tau_{\mathtt{p}^{-1}} = -\frac{\tau_{\mathtt{p}}}{\eta_{\mathtt{p}}}, \qquad \omega_{\mathtt{p}^{-1}} = \eta_{\mathtt{p}}^2 \omega_{\mathtt{p}}. \tag{3.11}$$

Following the transformations, the estimation is computed by a weighted average of those, that is controlled by the parameter ω. It is worth to notice that this parameter is equivalent to the inverse of error variance of the relational local generative model (3.1). Therefore, the estimate can be interpreted as the outcome of an aggregation with precision that ranks the relational information.

3.2 Multi-relational Propagation Algorithm

As we described initially, we target a node-level completion task where the multi-relational graph $\mathcal{G}$ is a priori given and the node states are known only at a subset of nodes $\mathcal{U} \subseteq \mathcal{V}$. Let us denote vector $\mathbf{x}$ storing the node values that we aim to solve for. We have observed values over $\mathcal{U}$, which are stored in another vector whose elements are $\{x_i^{(0)} : i \in \mathcal{U}\}$. Then, the graph regularization problem, which was previously stated in simple graphs (2.3) in Chapter 2, can be expressed in multi-relational settings as follows:

$$\min_{\mathbf{x}} \sum_{i \in \mathcal{V}} \sum_{\mathtt{p} \in \mathcal{P} \cup \mathcal{P}^{-1}} \sum_{\mathtt{r}(i,j)=\mathtt{p}} \omega_{\mathtt{p}} (x_i - \eta_{\mathtt{p}} x_j - \tau_{\mathtt{p}})^2 + \gamma \sum_{i \in \mathcal{U}} (x_i - x_i^{(0)})^2. \tag{3.12}$$

Instead of computing the closed form solution of this problem, which is computationally exhaustive in large scale settings, we follow the iterative framework suggested in [39]. As also discussed in Chapter 2, the label propagation algorithm is an iterative neighborhood aggregation method, where each iteration computes the solution of a first order Bayesian estimation problem on the graph. The first order Bayesian estimation yields an approximation of the node's state given its immediate neighbors, whereas the propagation algorithm expands the scope of this approximation at each iteration by processing the information originating from further neighborhoods. We also noted that these iterations converge to the solution of the graph regularization problem in (2.3).

In a similar manner, here, we propose a propagation algorithm that relies on the first order relational Bayesian estimate that is introduced in (3.3). The algorithm operates iteratively where the relational neighborhood aggregation (3.4) is accomplished at each node of the graph simultaneously. Thus, we denote a vector $\mathbf{x}^{(k)} \in \mathbb{R}^N$ composing the values at iteration-k over the set of nodes for $|\mathcal{V}| = N$. Next, we express the iterations in matrix-vector multiplication format.

Iterations in Matrix Notation. We first introduce matrix $\mathbf{A}_{\mathtt{p}}$ for encoding the adjacency pattern

of relation type p. Therefore, it is $(N \times N)$ asymmetric matrix storing the incoming edges on its rows and outgoing edges on its columns. Accordingly, one can compile the aggregations (3.4) accomplished simultaneously over the entire graph using a matrix notation. Then, the relational local operations at iteration-k can be expressed as follows:

$$\mathbf{x}^{(k)} = \left(\sum_{\mathrm{p}\in\mathcal{P}} \left(\omega_{\mathrm{p}} \left(\eta_{\mathrm{p}} \mathbf{A}_{\mathrm{p}} \mathbf{x}^{(k-1)} + \tau_{\mathrm{p}} \mathbf{A}_{\mathrm{p}} \mathbf{1} \right) + \omega_{\mathrm{p}} \eta_{\mathrm{p}} \left(\mathbf{A}_{\mathrm{p}}^{\top} \mathbf{x}^{(k-1)} - \tau_{\mathrm{p}} \mathbf{A}_{\mathrm{p}}^{\top} \mathbf{1} \right) \right) \right) \odot \left(\sum_{\mathrm{p}\in\mathcal{P}} \left(\omega_{\mathrm{p}} \mathbf{A}_{\mathrm{p}} \mathbf{1} + \omega_{\mathrm{p}} \eta_{\mathrm{p}}^{2} \mathbf{A}_{\mathrm{p}}^{\top} \mathbf{1} \right) \right)^{-1}, \quad (3.13)$$

where $\mathbf{1}$ is the vector of ones, $\odot$ stands for element-wise multiplication. In addition, the inversion on the latter sum term is applied element-wise. This part, in particular, arranges the denominator in Equation (3.4) in vector format. Thus, it can be seen as the normalization factor over the neighborhood aggregation. For the purpose of simplification, we re-write (3.13) as

$$\mathbf{x}^{(k)} = (\mathbf{T}\mathbf{x}^{(k-1)} + \mathbf{S}\mathbf{1}) \odot (\mathbf{H}\mathbf{1})^{-1}, \quad (3.14)$$

by introducing the auxiliary matrices

$$\mathbf{T} = \sum_{\mathrm{p}\in\mathcal{P}} \eta_{\mathrm{p}} \omega_{\mathrm{p}} (\mathbf{A}_{\mathrm{p}} + \mathbf{A}_{\mathrm{p}}^{\top}), \quad (3.15)$$

$$\mathbf{S} = \sum_{\mathrm{p}\in\mathcal{P}} \tau_{\mathrm{p}} \omega_{\mathrm{p}} (\mathbf{A}_{\mathrm{p}} - \eta_{\mathrm{p}} \mathbf{A}_{\mathrm{p}}^{\top}), \quad (3.16)$$

$$\mathbf{H} = \sum_{\mathrm{p}\in\mathcal{P}} \omega_{p} (\mathbf{A}_{\mathrm{p}} + \eta_{\mathrm{p}}^{2} \mathbf{A}_{\mathrm{p}}^{\top}). \quad (3.17)$$

Algorithm. Given the iterations above, we can now formalize the proposed algorithm that we call as Multi-relational Propagation (MRP). We introduce an indicator vector $\mathbf{u} \in \mathbb{R}^{N}$ which encodes initially known set of nodes and the propagated set of nodes throughout the iterations. Thus, it is initialized as $\mathbf{u}_{i}^{(0)} = 1$, if $i \in \mathcal{U}$, else 0. Then, the vector $\mathbf{x}$ stores the node values throughout the iterations. It is initialized by the values over $\mathcal{U}$, and, it is zero-padded at the unknowns, i.e., $\mathbf{x}_{i}^{(0)} = 0$ if $i \in \mathcal{V} \setminus \mathcal{U}$.

Similar to the label propagation algorithm [51], our algorithm fundamentally consists of aggregation and normalization steps. In order to encompass the multi-relational transformation procedure during the aggregation, we formulate an iteration of MRP by the steps of aggregation, shift and normalization respectively. In addition, similar to the Page-rank algorithm [50], we employ a damping factor $\xi \in [0,1]$ in order to update the node's state by combining its value from the previous iteration.

We provide a pseudocode for MRP in Algorithm 1. Here, we reserve that the propagation parameters for each relation type, $\{\tau_{\mathrm{p}}, \eta_{\mathrm{p}}, \omega_{\mathrm{p}}\}$ are estimated in advance over the known set

Algorithm 1: MRP

Input: $\mathcal{U}, \{x_i | i \in \mathcal{U}\}, \{\mathbf{A}_\mathrm{p}, \tau_\mathrm{p}, \eta_\mathrm{p}, \omega_\mathrm{p}\}_\mathcal{P}$
Output: $\{x_i | i \in \mathcal{V} \setminus \mathcal{U}\}$
Initialization: $\mathbf{u}^0, \mathbf{x}^0, \mathbf{T}, \mathbf{S}, \mathbf{H}$
for $k = 1, 2, \cdots$ **do**
  Step 1. Aggregate: $\mathbf{z} = \mathbf{T}\mathbf{x}^{(k-1)}$
  Step 2. Shift: $\mathbf{z} = \mathbf{z} + \mathbf{S}\mathbf{u}^{(k-1)}$
  Step 3. Aggregate the normalization factors: $\mathbf{r} = \mathbf{H}\mathbf{u}^{(k-1)}$
  Step 4. Normalize: $\mathbf{z} = \mathbf{z} \odot \mathbf{r}^\dagger$ // $\dagger$ is for element-wise pseudo-inverse
  Step 5. Update values:

$$\mathbf{x}_i^{(k)} = \begin{cases} \mathbf{x}_i^{(k-1)}, & \text{if } \mathbf{r}_i = 0 \quad \text{// null info at neighbors} \\ \mathbf{z}_i, & \text{if } \mathbf{r}_i > 0, \mathbf{u}_i^{(k-1)} = 0 \quad \text{// null info at the node} \\ (1-\xi)\mathbf{x}_i^{(k-1)} + \xi \mathbf{z}_i, & \text{e.w.} (\mathbf{r}_i > 0, \mathbf{u}_i^{(k-1)} = 1) \end{cases}$$

  Step 6. Update propagated nodes: $\mathbf{u}^{(k)} = \mathbf{u}^{(k-1)}, \quad \mathbf{u}_i^{(k)} = 1$ if $\mathbf{r}_i > 0$
  Step 7. Clamp the known values: $\mathbf{x}_i^{(k)} = x_i, \forall i \in \mathcal{U}$
break if all($\mathbf{u}^{(k)}$) & all($\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)} < \varepsilon$)
$x_i = \mathbf{x}_i^{(k)}, \forall i \in \mathcal{X} \setminus \mathcal{U}$.

of nodes $\mathcal{U}$, as described in Section 3.1.2. Then, we provide them to the algorithm as input together with the adjacency matrices encoding the multi-relational, directed graph. Steps 1-4 in MRP are essentially responsible for the multi-relational neighborhood aggregation—aggregation, shift and normalization. Then at Step-5, the nodes' states are updated based on the collected information from the neighbors. Here, the first case handles null information aggregated from neighbors mainly because the neighbors are unknown and not propagated yet. In this case, we leave the node's state as it is. In a second case where the node's current state is unknown and not propagated yet, we directly set it to the aggregated value from the neighbors. Otherwise, we employ the damping ratio, ξ, to update the node's state, which adjusts the amount of trade-off between the neighborhood aggregation and the previous state of the node. Moreover, we distinguish whether the current state of an arbitrary node is unknown or not by using the indicator vector, $\mathbf{u}^{(k)}$, which keeps track of propagated nodes throughout the iterations. Hence, in Step 6, we update it as well. Finally, in Step 7, we clamp the values at the known set of nodes, which means we leave their states unchanged, simply because they store the governing information for completing the missing ones. The algorithm terminates when all the nodes are propagated and the difference between two consecutive iterations is under a certain threshold. Accordingly, the number of iterations is related to the choice of hyperparameter ξ and the stopping criterion.

Although the algorithm is formalized with matrix-vector multiplications, we exploit sparse relational structure of a multi-relational graph in the implementation of MRP. Thus, aggregation steps in Algorithm 1 require $2|\mathcal{E}|$ operations, then, normalization and update steps require $|\mathcal{V}|$ operations at each iteration. Therefore, MRP scales linearly with the number of edges in

the graph, similar to the standard label propagation algorithm LP. We finally note that setting $\tau_p = 0$, $\eta_p = 1$, $\omega_p = 1\ \forall p \in \mathcal{P}$ manually, MRP drops down to LP[II] as if we operate on a simple, homogeneous graph regardless of the relation types and directions.

3.3 Experiments

We now present a proof of the proposed multi-relational propagation method for node regression task on two applications. First, we test MRP in estimating weather measurements on a multi-relational and directed graph that connects the weather stations. Second, we evaluate the performance in predicting people's date of birth, where people are connected to each other on a social network composing different types relationships.

In the experiments, the damping factor is set as $\xi = 0.5$, then the threshold for terminating the iterations is fixed to 0.1% of the range of given values. Then, as evaluation metrics, we use root mean square error (RMSE), mean absolute percentage error (MAPE) and normalized RMSE (nRMSE) with respect to the range of groundtruth values. The evaluation metrics are calculated over the initially unknown set of nodes, which can be counted as test nodes. In the experiments, η parameter in MRP is left by default as 1 since we do not empirically observe a scale change over the relation types given by the datasets we work on. Then, we realize the estimation of parameters τ and ω for the relation types based on the observed set of node values as described in Section 3.1.2.

3.3.1 Multi-relational Estimation of Weather Measurements

We test our method on a meteorological dataset provided by MeteoSwiss, which compiles various types of weather measurements on 86 weather stations between years 1981-2010 [III]. In particular, we use yearly averages of weather measurements in our experiments.

Construction of multi-relational directed graph. To begin with, we prepare a multi-relational graph representation $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{P})$ of the weather stations, *i.e.*, $|\mathcal{V}| = 86$, where we relate them based on two types of relationships, *i.e.*, $|\mathcal{P}| = 2$. First, we connect weather stations based on geographical proximity. Thus, we insert an edge between a pair of stations if the Euclidean distance between their GPS coordinates is below a threshold, on which we acquire 372 edges. The geographical proximity leads to a symmetric (bi-directed) relationship. Second, we relate the weather stations based on the altitude proximity in a similar logic. However, this time we anticipate an asymmetric relationship where the direction of an edge indicates an altitude ascend between weather stations. For both of the relation types, we adjust the threshold for building connections such that there is not any disconnected node. Consequently, altitude

[II] The label propagation algorithm [51] was originally designed for completing categorical features across a simple, weighted graph. By leaving the parameters of MRP as default, we actually revise it to propagate continuous features and apply for the node regression task.

[III] https://github.com/bayrameda/MaskLearning/tree/master/MeteoSwiss

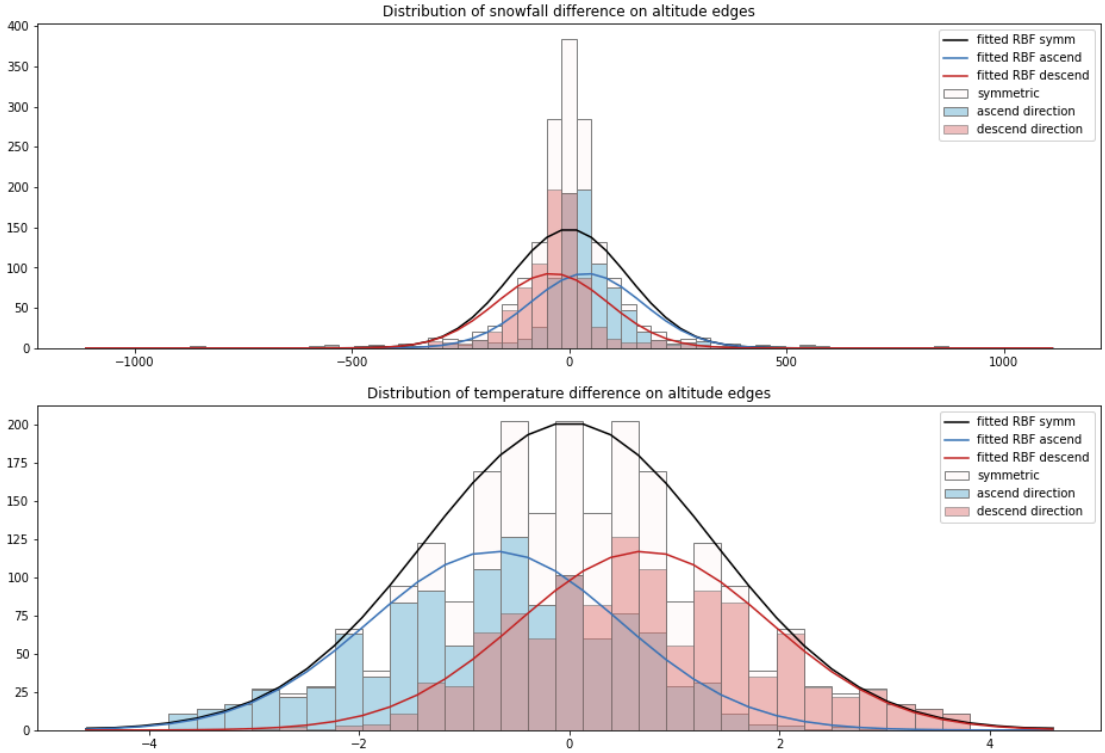


Figure 3.2 – Distribution of change in temperature and snowfall(cm) measurements between the weather stations that are related via altitude proximity. Differences are shown along the ascend and descend direction separately, then, symmetric distribution shows the changes regardless of the direction. Also, a radial basis function (RBF) is fitted to each histogram.

relations end up with 1144 edges.

In the experiments, we randomly sample initially known set of nodes, $\mathcal{U}$, from the entire node set, $\mathcal{V}$, with a ratio of 80%. The prediction performance are computed over $\mathcal{V} \setminus \mathcal{U}$. Then, we repeat the experiment in this setting for 50 times in Monte Carlo fashion. The evaluation metrics are then averaged over the series of simulations.

Table 3.2 – Temperature and Snowfall Prediction Performances

		RMSE	**MAPE**	**nRMSE**
Temperature	LP	1.120	0.155	0.050
	MRP	1.040	0.147	0.045
Snowfall	LP	194.49	0.405	0.112
	MRP	180.10	0.357	0.105

Predicting Temperature and Snowfall on Directed Altitude Graph

We first conduct experiments on a simple scenario where we target predicting temperature and snowfall measurements by MRP, which permits reasoning over the directed altitude relations. Hence, we compare the proposed method to the standard label propagation algorithm, LP, which overlooks asymmetric relational reasoning. In this regard, we aim at evaluating the importance of the directed transformation during the neighborhood aggregation that is mainly gained by the shift parameter, τ. In fact, this parameter directly corresponds to the mean of differences computed along the direction of the altitude edges—since $\eta = 1$. Then, the parameter ω is simply associated with the inverse of the variance of the differences. This can be visualized by fitted RBFs on the distribution of the measurement changes on the edges, which is shown in Figure 3.2. Here, we see that the temperature differences in the ascend direction, *i.e.*, $\{(x_i - x_j)\ \forall \mathtt{r}(i,j) = \mathtt{altitude\_ascend}\}$, has a mean in the negative region. This can be interpreted as an expected decrease in temperature values along altitude ascend. On the contrary, the mean of snowfall differences along the ascend direction has a positive value, which signifies a increase in snowfall as altitude rises.

As seen in Table 3.2, even in the case of single relation type—altitude proximity, incorporating the directionality in the graph and exploiting this with our propagation model MRP, we manage to record an enhancement in predictions over the regression realized by the label propagation, LP.

Predicting Precipitation on Directed, Multi-relational Graph

We now test our method in a further scenario where we integrate both altitude and geographical proximity relations to predict precipitation measurements on the weather stations. Figure 3.3 shows the distribution of the differences over both relation types. We see that along the direction of altitude edges, precipitation changes less asymmetrically compared to the differences captured in temperature and snowfall in Figure 3.2. In addition, while the variance over the GPS relations is calculated as 23.4×10^4, it is 18.7×10^4 over the altitude edges, which are inversely proportional to their parameter ω in MRP.

The prediction performance is compared to the regression by LP, that is accomplished over the altitude relations and GPS relations separately. Since MRP handles both of the relation types and the direction of the edges simultaneously, it achieves a better performance than LP, as seen in Table 3.3.

Table 3.3 – Precipitation Prediction Performances

	RMSE	**MAPE**	**nRMSE**
LP-altitude	381.86	0.261	0.174
LP-gps	374.38	0.242	0.168
MRP	347.98	0.238	0.157

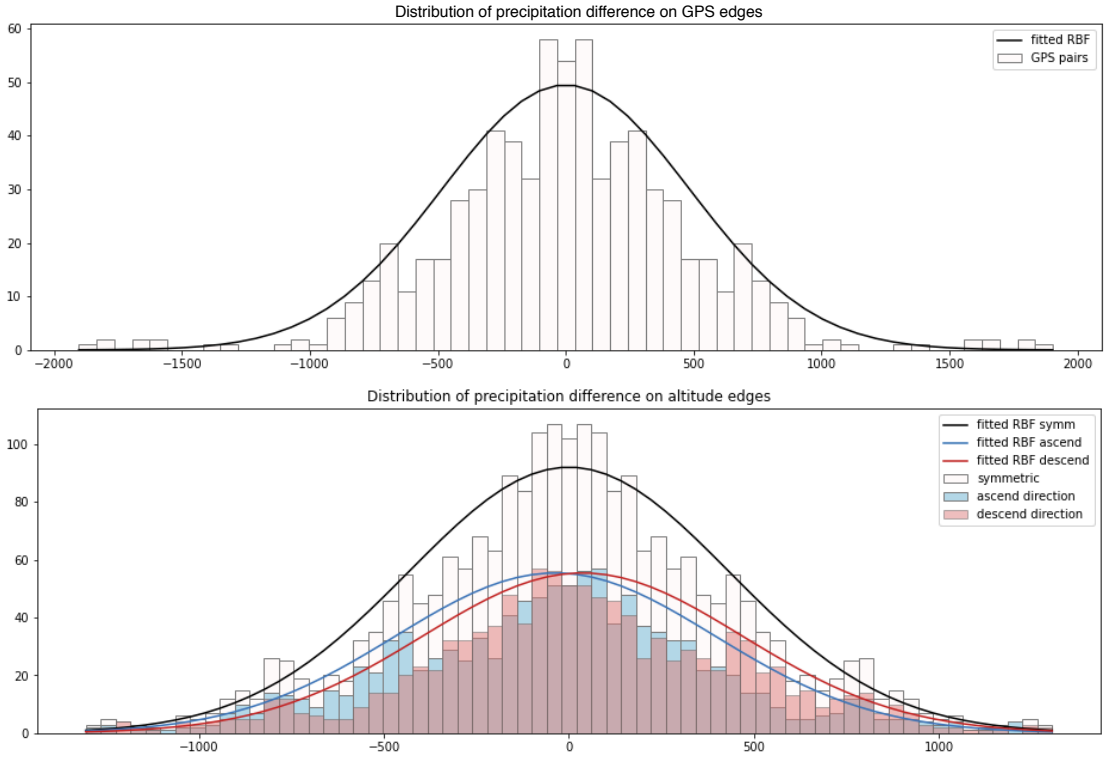


Figure 3.3 – Distribution of change in precipitation(mm) measurements between the weather stations that are related via geographical and altitude proximity.

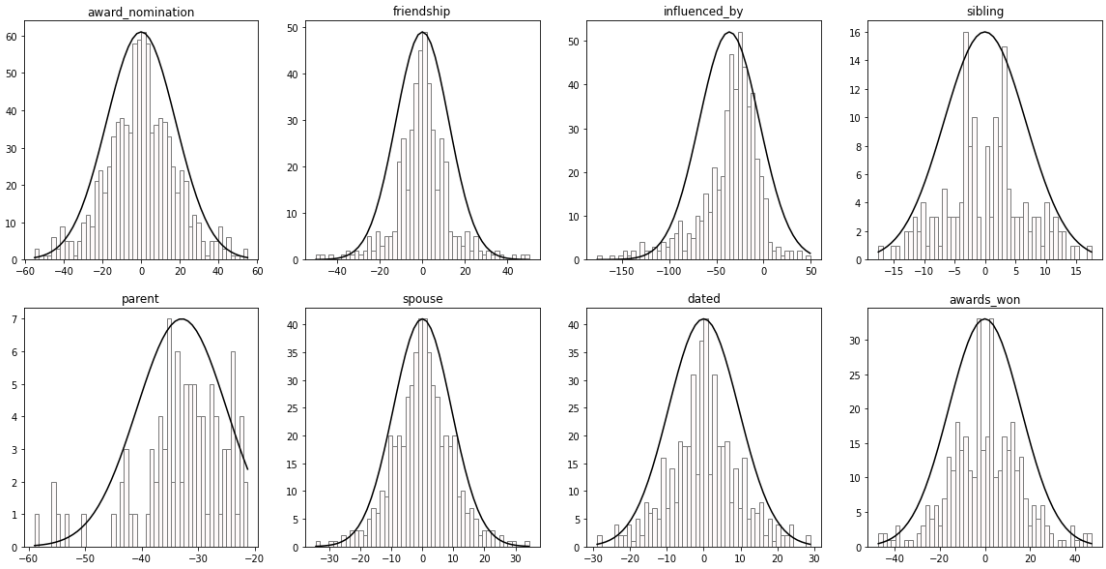


Figure 3.4 – Distribution of difference (year) in date of births over different types of relations between people.

3.3.2 Predicting People's Date of Birth in a Social Network

We also conduct experiment on a small subset of a relational database called Freebase [88]. For this purpose, we work on a graph $\mathcal{G}(\mathcal{V},\mathcal{E},\mathcal{P})$ composing 830 people, *i.e.*, $|\mathcal{V}| = 830$, connected

Relationship	edges	mean	variance
award_nomination	454	0	320.23
friendship	221	0	155.82
influenced_by	528	-36.25	1019.77
sibling	83	0	45.16
parent	98	-32.90	62.90
spouse	262	0	87.60
dated	231	0	90.95
awards_won	183	0	257.45

Table 3.4 – Statistics for each type of relation. Columns respectively: number of edges, mean and variance of the date of birth difference belonging to the associated relation type.

Table 3.5 – Date of Birth Prediction Performances

		RMSE	**MAPE**	**nRMSE**
LP	award_nomination	32.43	0.011	0.115
	friendship	31.92	0.011	0.113
	influenced_by	30.29	0.012	0.108
	sibling	32.69	0.012	0.116
	parent	33.62	0.013	0.119
	spouse	31.45	0.011	0.112
	dated	31.70	0.011	0.113
	awards_won	33.04	0.012	0.117
	union	24.22	0.008	0.086
MRP		15.62	0.005	0.055

via 8 different types of relationship, *i.e.*, $|\mathcal{P}| = 8$. Table 3.4 summarizes the statistics for each of them. Here, the task is to predict people's date of birth while it is only known for a subset of people. A fragment of the multi-relational graph is also illustrated in Fig. 3.1, where it can be seen that there are basically two types of asymmetric relations: `influenced_by` and `parent`. Thus, the direction of the edges are specifically significant for those. Such asymmetry is also shown by visualizing the distribution of the difference in date of births, which is given over each type of relationship in Figure 3.4. We note that here we try to fit a radial basis function to the histogram of the differences since the residual term in the local generative model (3.1) is assumed to be normally distributed.

In the experiments, we randomly select the set of people whose date of birth is initially known, $\mathcal{U}$ [IV], with a ratio of 50% in $\mathcal{V}$. We again report the evaluation metrics that are averaged over a

[IV] The performance comparisons are reported by averaging the evaluation metrics over a series of experiments where $\mathcal{U}$ is sampled at random with the aforementioned sparsity levels. The experiments are also conducted in setups with different sparsity of observed features. As expected, performance of the competitor algorithms enhances with larger set of observed features while the comparison between them with the reported results remains to be representative.

series of experiments repeated for 50 times.

We compare the performance of MRP to the regression of date of birth values obtained with label propagation LP. We run LP over the edges of each relation type separately and also at the union of those. The results are given in Table 3.5. Based on the results, we can say that the most successful relation types for predicting the date of birth seems to be `influenced_by` and `spouse` using LP. Nonetheless, when LP operates on the union of the edges provided by different type of relationships, it performs better than any single type. Moreover, MRP is able to surpass this record by enabling a smart neighborhood aggregation over different types of relations. Once again, we argue that its success is due to the fact that it regards asymmetric relationships, here encountered as `influenced_by` and `parent`. In addition, it assigns different level of importance to the predictions collected through different type of relationships based on the uncertainty estimated over the observed data.

3.4 Conclusion

In this chapter, we proposed MRP, a sophisticated version of label propagation algorithm for multi-relational and directed graphs and we show its superior performance on the node regression task. Although we here target imputing continuous values at the nodes of a multi-relational and directed graph, it is possible to generalize the proposed approach for node embedding learning and then for the node classification tasks. The augmentation of the computational graph of the propagation algorithm using multiple types of directed relationships provided by the domain knowledge permits anisotropic operations on graph, which is claimed to be promising for future directions in graph representation learning [62].

Moreover, the proposed relational neighborhood aggregation method hints a message passing framework that can operate over different type of edges and edge directions. In the next chapter, we focus on this utility while we aim at regression of heterogeneous node features, which brings the level of complexity one step further.

4 Heterogeneous Message Passing in Knowledge Graphs

The existing literature on knowledge graph (KG) completion mostly focuses on the link prediction task. However, knowledge graphs have an additional incompleteness problem: their nodes possess attributes, whose values are often missing. In this chapter, we address the numerical node attribute completion task in KGs. In the previous chapter, we have introduced the multi-relational propagation algorithm MRP for node regression. Here, we extend this approach in order to impute missing heterogeneous features at the nodes of a KG. We denote our novel algorithm as Multi-relational Attribute Propagation, MRAP. It employs a set of message functions in order to predict one node attribute from another depending on the relationship between the nodes and also the type of the attributes. The propagation mechanism operates iteratively in a message passing scheme that collects predictions at every iteration and updates the value of the node attributes. Similar to MRP, the parameters of MRAP are estimated over the observed set of node attributes prior to the iterative message passing scheme. However, it is possible to infer the parameters via back propagation within a semi-supervised learning scheme. Accordingly, we introduce an alternative end-to-end learning framework for node attribute completion and present a discussion over both frameworks. We conduct experiments over two benchmark datasets, which shows the effectiveness of the proposed approaches.

This chapter is organized as follows. We first introduce the task of completing numerical features in a KG, present related works and summarize our contribution. Then, we establish the notation used throughout the chapter, formulate the problem and propose the two alternative schemes for solution. Finally, we give the experimental results and the performance analysis of the proposed frameworks and conclude.

A part of this chapter is based on a joint work with Alberto Garcia-Duran and Robert West, titled: “Node Attribute Completion in Knowledge Graphs with Multi-Relational Propagation" [37].

4.1 Completion of Numerical Node Attributes in Knowledge Graphs

Knowledge graphs (KGs) have the capability of storing rich structural information consisting of multiple types of semantic entities connected by different types of relationships. In the last years, this has led to immense attention on knowledge graph completion methods, which aim at inferring missing facts in a KG by reasoning about the observed facts. Knowledge graph embedding methods are at the core of this progress, by learning latent representations for both entities and relations in a KG [31, 89]. In relational representation learning, graph neural network (GNN) [57] and message passing neural network (MPNN) [4] methods have also been effectively used. While originally these methods were designed for simple undirected graphs, there are also works that incorporate multi-relational information [66, 90, 67]. Despite the very large number of relational reasoning methods, these works have mostly addressed link prediction and node/graph classification problems. While these methods always harness features learned from the relational structure of the graph, they very often overlook other information such as the numerical properties of the entities. In this work, we shift the focus away from the aforementioned problems, and study the much less explored problem of node attribute prediction in KGs.

The node attribute prediction problem is especially challenging because of bewildering heterogeneity of the KG data. To begin with, each type of entity in a KG is usually entitled with a different set of attributes. Then, each type of these attributes, in general, is expressed in its own feature space, which compels a regression over a heterogeneous feature space. Moreover, the entities are connected to each other via different types of relationships, which promotes various types of dependencies between their possessed attributes. The relational structure provides very rich predictive information, thus, node attribute completion in KGs requires an abundant relational reasoning process. In this study, we particularly address the incompleteness in the numerical node attributes that are expressed in continuous values. Figure 4.1 depicts an example: the node New York does not have a value for its two numerical attributes, `latitude` and `area`. Similarly, we observe missing values in some attributes of other nodes of the KG. Node attribute completion is the task of finding appropriate values for the nodes' numerical attributes that do not have an annotated value.

Different to the existing approaches in KG completion, in node attribute completion task, we harness not only the relational structure of the graph, but also the correlation between various types of node attributes. Humans also use these inputs to perform numerical reasoning. For instance, in Figure 4.1, one may provide an estimate about the date of death of Francis Ford Coppola by looking at the release date of one of his most popular movies. Accordingly, in this study, we impute the values of missing attributes by propagating information across the multi-relational structure of the KG. Our numerical reasoning also depends on the correlation between various types of attributes observed at the neighboring nodes. Thus, we design the propagation algorithm in a way that it operates by exchanging messages between source and target node attributes through the relationship between the nodes accommodating them. Type of a certain message is determined by its source attribute type, target attribute type

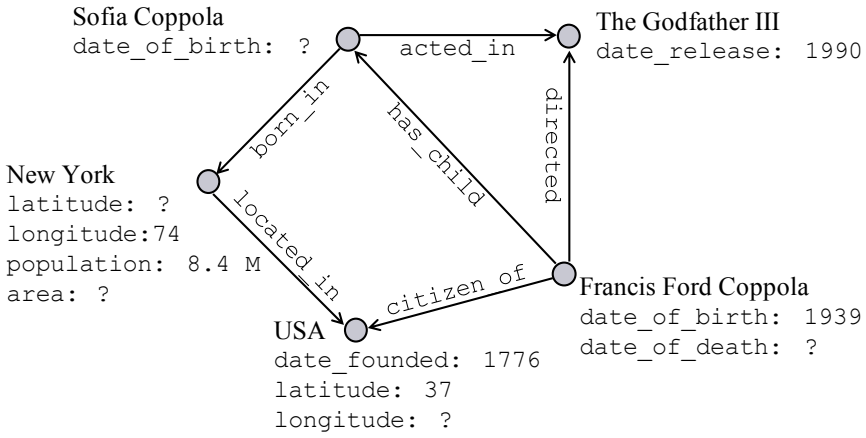


Figure 4.1 – A part of KG data with incomplete node attributes

and the relation type. Consequently, we employ a number of message functions that predict an attribute of a node from an attribute of its neighbor with respect to both the type of the attributes and the relation between the nodes. We also adopt another set of message functions for pair of attributes that accommodate at the same node, for instance, predicting the date of death of Francis Ford Coppola from his own date of birth. In addition, humans have the capacity to determine the predictive power of each source of information, and weight each information accordingly in their numerical reasoning process. Similarly, we assign a weight to each message function reflecting its predictive power, which will be taken into account during the aggregation of their messages.

For the proposed multi-relational propagation algorithm MRAP, the parameters of the message functions and their weights are estimated based on the observed set of node attributes prior to the propagation procedure. In addition, we propose an alternative, end-to-end, semi-supervised learning scheme where we infer the propagation parameters through a back-propagation procedure.

Related Work. Although many KGs often contain numerical properties attributed to the entities, very few studies have explored and exploited them [91, 92]. The numerical attribute prediction problem, in particular, was recently introduced by Kotnis and Garcia-Duran [93], who address the problem with a two-step framework called NAP++. First, they extend the KG embedding method to learn node embeddings underlying a KG enriched with numerical node attributes. Second, they build a k-NN graph upon the embedding to propagate the known values of node attributes towards the missing ones. Propagating information on a surrogate graph constructed on the embedding is rather sub-optimal compared to leveraging the original relational structure of the KG. As opposed to that, in this study, we propose a propagation algorithm that directly operates on the inherent structure of the KG. For this purpose, we take inspiration from the well-known label propagation algorithm [51], which infers the label of a node from its neighbors iteratively under the assumption that nearby nodes should have

similar values—we refer the reader to Section 2.1.2 for a revisit. However, this technique is insufficient to handle the complexity of KGs, which possess multiple types of attributes and multiple types of relationships following different affinity rules between neighboring nodes. For example, two nodes linked via the relationship `has_child` exhibit a certain bias between their `date_of_birth` attributes, but do not necessarily have similar values. The authors in [91] exploit such numerical node attributes in a KG for the multi-relational link prediction task. Instead of straightforwardly adopting the Euclidean distance between the neighboring node attributes, they model the affinity using a radial basis function, which can account for the aforementioned bias term that may arise in some relations. In our method, we introduce message functions modeling a linear relation between neighboring node attributes, which account for both any scale change between their feature spaces and also the bias between them. The key insight of our propagation model is that it has the capability of capturing the linear dependency between different types of attributes over different types of relationships. Therefore, our method allows message passing not only between node attributes of the same type but also between different types, unlike the previous numerical attribute propagation solution [93].

The GNN and MPNN methods also learn node representations by propagating them along the edges of a graph. Recently, multi-relational variants have also been developed, which usually augment the learning parameters in a relation-specific manner [58, 66, 94, 95, 69, 68, 96, 67]. Here, we refer the reader to Section 2.1.2 where we review the neural network schemes on simple graphs and their multi-relational variants. GNNs have also been studied on knowledge graphs in [97, 92, 90, 98]. Later, attention mechanism is adapted on multi-relational and heterogeneous graphs [99, 100, 101, 70, 71]. Attention enables discriminating the importance of the neighboring nodes for the inference task, rather than treating them equally. Similar to an attention mechanism, in our method, the weight assigned for a certain message type captures the importance of the collected predictions for its target node attribute. On the other hand, an important technical difference of our approach from the aforementioned GNN studies is that our method propagates incomplete node features across the graph instead of propagating fixed dimension of node representation vectors—embeddings. In graph representation learning studies, the embedding vector typically consists of a fixed set of node features. In our case, however, we do not have a fixed dimension of node feature vector, where the number of attributes assigned to each node varies. Thus, we choose to regress one existing node feature from another in a pairwise manner.

The graph representation learning literature is pretty centered around the works on node classification and the link prediction tasks, whereas very few studies address the node regression task. In particular, the node attribute completion on simple graphs has been studied previously by the authors in [102]. They work on attribute-missing graphs which entitles: the features attributed to a particular set of nodes are entirely missing. According to their categorization, our work focuses on attribute-incomplete graphs where the set of features attributed to the nodes are partially missing. Also, while they target completing either numerical or categorical node features, we aim at completing heterogeneous numerical features.

Furthermore, heterogeneous node regression is studied by the authors in [86]. While they target regression of fixed set of heterogeneous features on simple graphs, we aim at regression of varying set of heterogeneous features at the nodes of a multi-relational graph.

Contributions. In this study, we propose a multi-relational propagation algorithm, MRAP, which directly operates on the original structure of the knowledge graph. MRAP imputes missing numerical attributes by iteratively applying two steps to each node attribute: it collects all predictions about the node attribute and updates its value by aggregating the predictions based on their weights. We formulate MRAP within a message passing scheme reframed in [65]. Developing on top of MRAP framework, we also propose an alternative semi-supervised learning scheme which infers the node attributes and learn the propagation parameters in an end-to-end fashion. To the best of our knowledge, we are the first one to realize message passing with incomplete heterogeneous node features and demonstrate its applicability for the node attribute completion task. The message functions employed in propagation are interpretable in the sense that they capture a linear dependency between various types of node attributes through different types of relationships. The associated weights with them capture their predictive power, which then leads the aggregation of messages similar to an attention mechanism. Our proposed solutions for the node attribute completion problem are computationally cheaper than embedding learning approaches.

4.2 Multi-Relational Attribute Propagation

Notation. A KG enriched with node attributes is denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{P}, \mathcal{A})$, where $\mathcal{V}$ is the set of nodes (entities), $\mathcal{P}$ is the set of relation types, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{P} \times \mathcal{V}$ is the set of multi-relational edges, and $\mathcal{A}$ is the set of attribute types. Moreover, $\mathcal{N}_v$ is the set of all neighbors of node $v \in \mathcal{V}$, and $\mathcal{A}_v$ is the set of attributes belonging to v. The function $\mathtt{r}(v,n)$ returns the relation type $\mathtt{p} \in \mathcal{P}$ that is pointed from node n to node v. If such a relation exists between them, yet pointed from the node v to the node n, then the function returns the reverse as $\mathtt{p}^{-1}$. In addition, we denote x_n for the value of attribute x belonging to node n, *i.e.*, $x \in \mathcal{A}_n$.

4.2.1 Heterogeneous Local Generative Model for Numerical Attributes

We first introduce a local generative model for heterogeneous numerical features attributed to the nodes of a KG. We recall that in Chapter 3, we introduced a local generative model for the node features of a unique type accommodated on a multi-relational and directed graph. Here, we upgrade that model in order to conform the heterogeneous node attributes of multiple types. Thus, we also model the relationship between different types of node features that are attributed to neighboring nodes or to the same node as follows:

$$y_v = \begin{cases} \eta_{\mathtt{p}}^{y|x} x_n + \tau_{\mathtt{p}}^{y|x} + \epsilon, & \forall n \neq v,\ \mathtt{r}(v,n) = \mathtt{p} \text{ where } \epsilon \sim \mathcal{N}(0, (\sigma_{\mathtt{p}}^{y|x})^2) \\ \eta^{y|x} x_v + \tau^{y|x} + \epsilon, & \forall x \neq y \text{ where } \epsilon \sim \mathcal{N}(0, (\sigma^{y|x})^2), \end{cases} \tag{4.1}$$

where the first case builds the dependency of an attribute of type y on an attribute of type x through a relation of type p that holds between the nodes accommodating y and x—nodes v and n respectively. On the other hand, the second case models the dependency of attribute y on an attribute x accommodating at the same node—node v.

We empirically observed that such linear dependency holds very often between the attributes found in knowledge bases such as DBpedia or Freebase. For instance, `date_of_birth` of a person type of entity can be estimated through a certain value difference from that of a neighbor connected via the relation type `has_child`. This motivates the usage of the bias parameter τ. On the other hand, the attributes can be expressed in different units or ranges, for instance, `weight` of a node can be regressed with a linear correlation to its `height`, which motivates the parameter η.

Heterogeneous Message Functions. The local generative model given in (4.1) models a linear relationship between the dependent and independent heterogeneous node attributes. Following that, we introduce a number of functions which facilitate a shortcut for the information exchange between these source and target node attributes. These message functions are specific to the source and target attribute types, y and x respectively. Depending on the first case in (4.1), we denote function $f^{y|x}_{\mathtt{r}(v,n)} : \mathbb{R} \rightarrow \mathbb{R}$ to be applied to an explanatory variable x_n where the independent attribute x appears at a neighboring node n connected by the relation $\mathtt{r}(v,n)$:

$$f^{y|x}_{\mathtt{r}(v,n)}(x_n) = \eta^{y|x}_{\mathtt{r}(v,n)} x_n + \tau^{y|x}_{\mathtt{r}(v,n)}. \tag{4.2}$$

Then, depending on the second case in (4.1), we denote function $f^{y|x} : \mathbb{R} \rightarrow \mathbb{R}$ to be applied to another attribute x than the dependent attribute type, y, encountered at the same node v:

$$f^{y|x}(x_v) = \eta^{y|x} x_v + \tau^{y|x}, \tag{4.3}$$

which is obviously relation independent.

We now draw attention to the fact that the introduced message functions actually generate different types of messages regarding the type of source and target attribute and the relation type between the nodes accommodating them. For instance, it is possible to summarize them using the following notation:

$$\{ < x, \mathrm{p}, y > \mid x, y \in \mathcal{A}, \mathrm{p} \in \mathcal{P} \} \subseteq \mathcal{A} \times \mathcal{P} \times \mathcal{A}.$$

In addition, for the message types exchanged within the same node are summarized as:

$$\{ < x, y > \mid x, y \in \mathcal{A}, x \neq y \} \subseteq \mathcal{A} \times \mathcal{A}.$$

First-Order Estimate. We now adopt the introduced heterogeneous local generative model in order to derive an estimate of a node attribute in a KG. Here, we refer to previous chapters

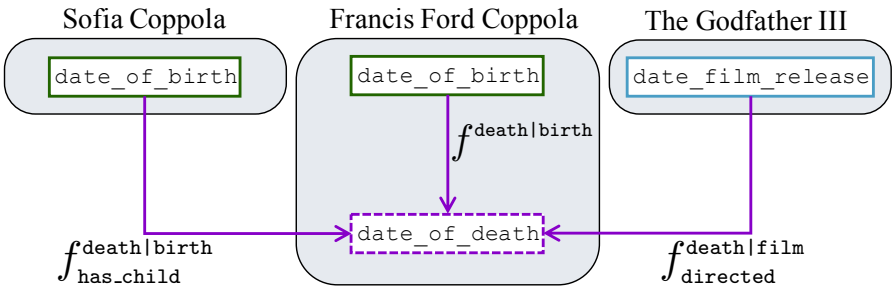


Figure 4.2 – Message passing performed for updating the attribute `date_of_death` for the node Francis Ford Coppola.

where we derive the Bayesian estimate of a node's state from its first-order neighbors. We remind that this problem is established by maximum a posteriori estimation of the node's value as formulated in (2.12). Due to the settings considered, it follows with the minimization of the negative log-likelihood, see (2.14) and (2.15). Particularly in Chapter 3, we formulate such a problem in terms of first-order multi-relational neighbors (3.3). Here, we consider similar settings, thus we follow the same steps. Consequently, we propose to solve the following problem for an approximation of a node attribute in a KG in terms of other types of attributes encountered at the same node and the attributes encountered at the first-order neighboring nodes.

Problem 1: Estimation of the node's attribute with heterogeneous local neighborhood

$$\underset{y_v}{\operatorname{argmin}} \quad \Bigg(\underbrace{\sum_{n \in \mathcal{N}_v} \sum_{x \in \mathcal{A}_n} \omega^{y|x}_{\mathbf{r}(v,n)} \big(y_v - f^{y|x}_{\mathbf{r}(v,n)}(x_n)\big)^2}_{\text{outer loss}} + \underbrace{\sum_{x \in \mathcal{A}_v, x \neq y} \omega^{y|x} \big(y_v - f^{y|x}(x_v)\big)^2}_{\text{inner loss}} \Bigg). \tag{4.4}$$

Since the heterogeneous local generative model (4.1) leads to Gaussian likelihood function, minimizing the negative log-likelihoods lead to the objective 4.4, where the message functions 4.2 and 4.3 are already plugged in. The loss emerges as the Euclidean distance between the node attribute to be estimated, y_v, and the values yielded by the message functions that are applied on the attributes at the first order neighbors, $\mathcal{N}_v$, and the central node, v. The outer loss is led by the attributes at the neighboring nodes depending on the first case of (4.1), whereas, the inner loss is led by the attributes within the same node depending on the second case of (4.1). At each case, the squared distances are in multiplication with the inverse of the associated error variances in (4.1), which are denoted by the parameters $\omega^{y|x}_{\mathrm{p}} = (\sigma^{y|x}_{\mathrm{p}})^{-2}$ and $\omega^{y|x} = (\sigma^{y|x})^{-2}$. This means the loss leads to a least squares problem. If we denote it by $\mathcal{L}^{y}_{v}$,

then, its solution can be found as $\frac{\partial \mathcal{L}_v^y}{\partial y_v}(\hat{y}_v) = 0$:

$$\hat{y}_v = \frac{\sum_{n \in \mathcal{N}_v} \sum_{x \in \mathcal{A}_n} \omega^{y|x}_{\mathtt{r}(v,n)} f^{y|x}_{\mathtt{r}(v,n)}(x_n) + \sum_{\substack{x \in \mathcal{A}_v \\ x \neq y}} \omega^{y|x} f^{y|x}(x_v)}{\sum_{n \in \mathcal{N}_v} \sum_{x \in \mathcal{A}_n} \omega^{y|x}_{\mathtt{r}(v,n)} + \sum_{\substack{x \in \mathcal{A}_v \\ x \neq y}} \omega^{y|x}}. \tag{4.5}$$

We draw attention to the fact that the estimate is simply obtained as a weighted and normalized sum of the transformations yielded by the message functions. We can interpret this as if each message function has an associated weight parameter $\omega^{y|x}_{\mathtt{p}}$ (or $\omega^{y|x}$). In addition, the message functions are linear regression functions depending on the local generative model (4.1), and the associated weight parameters are the inverse of the error variance of the regression models. This can be interpreted as if the weights are reflecting predictive power of the messages since they relate to the uncertainty of the regression models.

4.2.2 Algorithm MRAP

Ultimately, our learning objective is formalized as the minimization of the loss for each attribute belonging to each node of the graph *i.e.*, $\sum_{v \in \mathcal{V}} \sum_{y \in \mathcal{A}_v} \mathcal{L}_v^y$. In order to converge to its solution, we propose a propagation algorithm denoted by MRAP that operates iteratively. At each iteration, for each node v and each of its numerical attribute y, we aggregate all messages that aim at predicting y_v. This aggregation is realized based on the estimate derived in (4.5) where the contribution of each message is controlled by its corresponding weight. Here, the denominator is a normalization factor, *i.e.*, sum of the weights of the collected messages. For an illustration, we refer to Figure 4.2 where the messages are collected by the node Francis Ford Coppola that predicts his `date_of_death` attribute.

Next, we update the value of y_v using the aggregated messages. For this purpose, at iteration-k, the new estimate, $\hat{y}_v$, is combined with the previous value of the node attribute, y_v^{k-1}, via a damping factor $\xi \in (0,1)$ as follows

$$y_v^k = (1-\xi)\, y_v^{k-1} + \xi\, \hat{y}_v. \tag{4.6}$$

We design MRAP proceeding in steps of aggregation and update that are repeated for a certain number of iterations or until a convergence threshold is reached. Thus, the proposed method recovers the values of missing node attributes by minimizing their distances to the messages collected from such internal and external sources of information based on their weights. At each iteration, while the values of all missing attributes are updated, the values of a priori known attributes are clamped.

MRAP can be framed within a message passing algorithm. The framework defines two generic functions. The function AGGREGATE collects all messages targeted for a node attribute and

aggregates them. The function COMBINE takes the aggregation and the previous state to output a new state. In our approach, functions AGGREGATE and COMBINE correspond to Eq. (4.5) and (4.6), respectively. MRAP is given in Algorithm 2 using the message passing terminology. MRAP fuses different types of messages across the KG as mentioned previously. Accordingly, the pathways of the collected messages compose a computational graph that is an augmented version of the given structure of the KG:

$$\big\{(x_n, y_v) \quad \forall \mathtt{r}(v,n) \in \mathcal{P}\big\} \cup \big\{(x_v, y_v) \quad \forall\, x \neq y\big\}.$$

Algorithm 2: MRAP

Input: $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{P}, \mathcal{A})$, message functions with their associated weights
Output: Imputed node attributes
Initialization: $x_n^0 = x_n$ for a priori known attributes
for *Until Convergence* **do**
  for $y \in \mathcal{A}_v, \forall\, v \in \mathcal{V}$ **do**
    $\hat{y}_v = \text{AGGREGATE}(\{x_n^{k-1} |^{n \in \mathcal{N}_v,}_{x \in \mathcal{A}_n}\} \cup \{x_v^{k-1} |^{x \in \mathcal{A}_v}_{x \neq y}\})$
    $y_v^k = \text{COMBINE}\big(y_v^{k-1}, \hat{y}_v\big)$
  Clamp a priori known node attributes

Estimation of Parameters

MRAP explicitly makes use of the multi-relational structure given by the KG and the observed numerical node attributes to infer the missing ones. While it imputes the missing node attributes by iteratively applying Eq. (4.5) and (4.6), the message functions and their associated weights are computed in advanced, and kept fixed during the propagation process.

We obtain the propagation parameters by following the same steps as in Section 3.1.2. In MRAP, the heterogeneous message functions and their associated weights are originated from linear regression models—the heterogeneous local generative model in (4.1). It is possible to derive the parameters of a simple linear regression model from the samples of the dependent and independent variables. Thus, the parameters of the message functions (4.2), (4.3) are estimated from the observed set of node attributes.

Let $\mathcal{E}_{\mathtt{p}}^{(y,x)}$ be the set of pairs of nodes (v,n) where the relation type $\mathtt{p}$ is pointed from node n to node v, and for which the attributes y and x are observed in nodes v and n, respectively. We estimate the parameters of the regression function $f_{\mathtt{p}}^{y|x}$ as follows:

$$\eta_{\mathtt{p}}^{y|x} = \frac{\sum\limits_{(v,n) \in \mathcal{E}_{\mathtt{p}}^{(y,x)}} (y_v - \mu^y)(x_n - \mu^x)}{\sum\limits_{(v,n) \in \mathcal{E}_{\mathtt{p}}^{(y,x)}} (x_n - \mu^x)^2}, \tag{4.7}$$

where μ^x is the mean of attribute x. Consequently,

$$\tau_{\mathrm{p}}^{y|x} = \mathrm{mean}(\{(y_v - \eta_{\mathrm{p}}^{y|x} x_n) \,|\, (v,n) \in \mathcal{E}_{\mathrm{p}}^{(y,x)}\}), \tag{4.8}$$

Then, the error variance of the model is calculated as follows:

$$(\sigma_{\mathrm{p}}^{y|x})^2 = \mathrm{mean}(\{(y_v - \eta_{\mathrm{p}}^{y|x} x_n - \tau_{\mathrm{p}}^{y|x})^2 \,|\, (v,n) \in \mathcal{E}_{\mathrm{p}}^{(y,x)}\}). \tag{4.9}$$

In this way, we derive the parameters for constructing a message of type $< x, \mathrm{p}, y >$ where $f_{\mathrm{p}}^{y|x}$ applies. Now, suppose that we would like to predict x from y through the inverse relationship $\mathtt{r}(n,v) = \mathrm{p}^{-1}$. Then, we rewrite the local generative model by reversing the relation in the first case of (4.1):

$$x_n = \frac{1}{\eta_{\mathtt{r}(v,n)}^{y|x}} y_v - \frac{\tau_{\mathtt{r}(v,n)}^{y|x}}{\eta_{\mathtt{r}(v,n)}^{y|x}} - \frac{1}{\eta_{\mathtt{r}(v,n)}^{y|x}} \epsilon, \tag{4.10}$$

where the model parameters are diverted and the standard deviation of the error is rescaled by the factor of $\eta_{\mathtt{r}(v,n)}^{y|x}$. Accordingly, the parameters for constructing a message of type $< y, \mathrm{p}^{-1}, x >$, where $f_{\mathrm{p}^{-1}}^{x|y}$ applies, will correspond to:

$$\eta_{\mathrm{p}^{-1}}^{x|y} = \frac{1}{\eta_{\mathrm{p}}^{y|x}}, \quad \tau_{\mathrm{p}^{-1}}^{x|y} = \frac{-\tau_{\mathrm{p}}^{y|x}}{\eta_{\mathrm{p}}^{y|x}}, \quad w_{\mathrm{p}^{-1}}^{x|y} = \frac{(\eta_{\mathrm{p}}^{y|x})^2}{(\sigma_{\mathrm{p}}^{y|x})^2}. \tag{4.11}$$

Next, the parameters of the message functions of the inner loss in Problem 1, $f^{y|x}$, are computed by following a similar procedure. Let $\mathcal{V}^{(y,x)}$ denote the set of nodes for which both the attributes y and x are observed as y_v and x_v respectively. In Eq. (4.7), (4.8) and (4.9), we replace $\mathcal{E}_{\mathrm{p}}^{(y,x)}$ by $\mathcal{V}^{(y,x)}$ in order to estimate the parameters of the message function given in (4.3).

We finally note that if the linear dependency described in (4.1) does not exist for a certain type of message, it is possible to exclude those type of messages from MRAP. For this purpose, upon estimating the model parameters, one can check whether the normal error assumption is fulfilled or not.

4.3 Semi-supervised Learning Scheme

We now propose an alternative semi-supervised learning scheme which imputes the numerical node attributes in an end-to-end fashion. As it is explained, MRAP framework requires the propagation parameters to be estimated over the observed pair of node attributes and given to the algorithm. This means that the parameters are approximated over one hop connections between the observed attributes, *i.e.*, $\mathcal{E}_{\mathrm{p}}^{(y,x)}$, $\mathcal{V}^{(y,x)}$, although we expand the scope of the

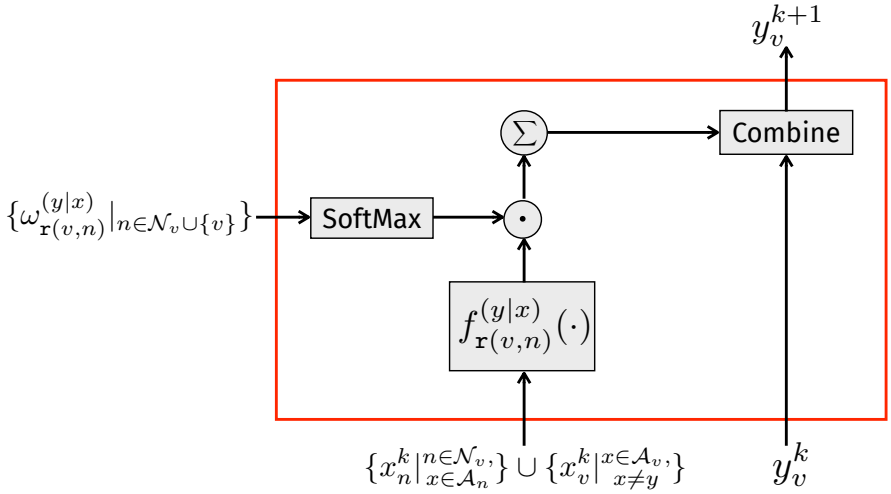


Figure 4.3 – Update of a node attribute in one iteration of forward propagation. For the sake of grouping the messages, we grant that $\mathtt{r}(v,v)$ returns null.

approximation of missing node attributes to the further hope neighborhoods at each iteration of the propagation. It is also possible to learn the parameters minimizing the loss that is calculated subsequent to the propagation procedure. For this purpose, we follow a forward-backward learning scheme described as follows.

Forward Propagation. The forward propagation operates similar to MRAP 2 with a slight difference. We launch the propagation algorithm with some default set of propagation parameters ($\tau = 0, \eta = 1$, and equal ω for each type of message), and excite the iterations with the observed set of attributes as usual. However, at the end of the iterations, we skip the clamping step in MRAP. Also, specifically after the first iteration, we unlabel the observed set of attributes, and treat them as if they were missing and to be completed. This means we do not re-inject the true values of the observed node attributes throughout the iterations of the forward propagation. Thus, we allow a residual error emerging at the observed node attributes between their true values and their estimated values at the end of the forward propagation. This residual error is associated with the propagation parameters where the forward propagation is conducted. We then repeat the forward propagation with an updated set of parameters which minimize the residual error.

An iteration of the forward propagation is illustrated in Figure 4.3 for the update of a certain node attribute. As seen, we do not use the weight parameters directly in the aggregation yet we apply them through a softmax function, which guarantees non-negative contribution of the collected messages, and also normalizes the contributions[I].

[I]There is no softmax function used in the algorithm MRAP because weight parameters of MRAP are estimated as the precision of the linear regression model of heterogeneous messages. These are already calculated as non-negative. However, in the semi-supervised learning scheme, we did not constrain the weight parameters to be learned to a non-negative search space. Instead, we utilize the softmax function for the set of weight parameters accounting for the aggregation step.

Back Propagation. As indicated, at the end of each forward pass, there is a loss generated on the observed attributes between their true values and the inferred values. Therefore, we update the propagation parameters—τ, η, ω for each type of message—by minimizing this loss using gradient descent, which is also called as back propagation. Through the epochs of such a forward-backward scheme, we infer both the propagation parameters and the missing node attributes simultaneously. Such an alternative inference scheme to MRAP can be particularly useful where the observed set of node attributes do not constitute sufficient number of pairs to estimate the propagation parameters in advanced.

Compositional Attention Mechanism. As indicated, the proposed semi-supervised learning scheme is parameterized for each message type. Thus, one should beware of a possible over-parameterization issue in case of very large number of message types induced by the heterogeneity of the input graph and the numerical features. Especially when the type of message paths across the computational graph is unevenly distributed, there could be few occurrences for certain types of messages. This can be resolved with certain parameter sharing or regularization strategies. In particular here, we mention a compositional attention mechanism in order to reparameterize the message weights. Message weights are specifically important since they play role in determining the contribution of the messages predicting an attribute. In other words, they can be interpreted as the attention coefficients assigned for the exchanged messages throughout the propagation. As explained, in the semi-supervised learning scheme, for message type $< x, \mathtt{p}, y >$, we learn a weight parameter $\omega_{\mathtt{p}}^{y|x}$. It is possible to decompose the weight to the parties of the associated message type as follows:

$$\omega_{\mathtt{p}}^{y|x} = \mathbf{h}_x^\intercal \text{diag}(\mathbf{h}_{\mathtt{p}}) \mathbf{h}_y, \tag{4.12}$$

where $\mathbf{h}_x, \mathbf{h}_{\mathtt{p}} \in \mathbb{R}^d$ are representation vectors for attribute type x and for relation type $\mathtt{p}$ respectively—for the inner node messages $\mathbf{h}_{\mathtt{p}}$ can be taken as vector of ones. Such an attention yields symmetric decomposition with respect to source and target attribute types. However, for heterogeneous message passing, asymmetric decomposition might be more useful. In that case, the following decomposition can be employed:

$$\omega_{\mathtt{p}}^{y|x} = \mathbf{h}_{\mathtt{p}}^\intercal [\mathbf{h}_x ; \mathbf{h}_y], \tag{4.13}$$

where $\mathbf{h}_{\mathtt{p}} \in \mathbb{R}^{2d}$.

All in all, we can summarize the message passing operation executed in the forward pass as follows:

$$y_v^{(k+1)} = \text{COMBINE}\Big(y_v^{(k)}, \text{AGGREGATE}\Big(\big\{ \text{ATTENTION}(f_{\mathtt{r}(v,n)}^{y|x}(x_n)),\ \forall x \in \mathcal{A}_n,\ n \in \mathcal{N}_v \cup \{v\} \big\} \Big) \Big), \tag{4.14}$$

where $\mathtt{r}(v, v)$ returns null. Then, AGGREGATE accomplishes the weighted sum of the collected

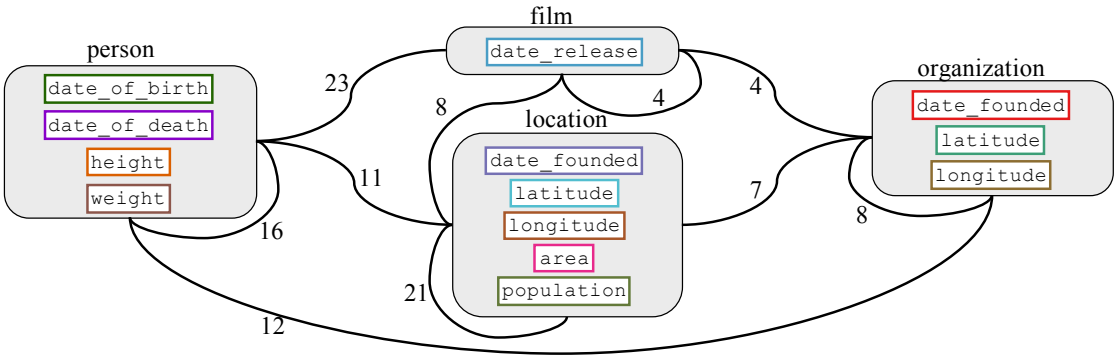


Figure 4.4 – A summary of FB15K-237 with entity types and numerical attributes encountered on them. The number attached to the connection between a pair of entity types indicates the number of relationship types between those entities.

messages as

$$\sum_{\substack{x\in\mathcal{A}_n,\\ n\in\mathcal{N}_v\cup\{v\}}} \alpha_{v|n}^{y|x} f_{\mathbf{r}(v,n)}^{y|x}(x_n)). \tag{4.15}$$

where the attention weights are applied with a softmax function in the aggregation:

$$\alpha_{v|n}^{y|x} = \frac{\exp\left(\omega_{\mathbf{r}(\mathrm{v,n})}^{\mathrm{y|x}}\right)}{\sum_{\substack{x\in\mathcal{A}_m,\\ m\in\mathcal{N}_v\cup\{v\}}} \exp\left(\omega_{\mathbf{r}(\mathrm{v,m})}^{\mathrm{y|x}}\right)} \tag{4.16}$$

4.4 Experiments

We evaluate the performance of the proposed frameworks on two KG datasets whose nodes have numerical attributes: FB15K-237 [88] and YAGO15K [103]. In order to illustrate the complexity of the data, we summarize FB15K-237 dataset in a diagram given in Figure 4.4 with the attribute types of interest in the experimental study and the types of entities accommodating those. The number of node attributes of each type encountered in each dataset are also listed in Table 4.1. Two error metrics are used to assess the performance: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), which are measured on each type of attribute individually.

4.4.1 Performance of MRAP

We implement MRAP[II] using the PyTorch-scatter package [104], which provides an efficient computation of message passing on a sparse relational structure. The damping factor of MRAP is set to $\xi = 0.5$, and the propagation stops upon reaching a convergence when the difference

[II] Source code is available at https://github.com/bayrameda/MrAP

Table 4.1 – Number of node attributes encountered in datasets for each attribute type. The upper block contains numerical attributes of date type. The lower block contains all other attributes. A dash (-) indicates the corresponding attribute is not encountered in the dataset.

Attribute	FB15K-237	YAGO15K
`date_of_birth`	4406	8217
`date_of_death`	1214	1821
`film_release`	1853	-
`organization_founded`	1228	-
`location_founded`	917	-
`date_created`	-	6574
`date_destroyed`	-	536
`date_happened`	-	388
`latitude`	3190	2989
`longitude`	3192	2989
`area`	2154	-
`population`	1920	-
`height`	2855	-
`weight`	225	-

between two consequent iterations drops below 0.1% of the range of attributes.

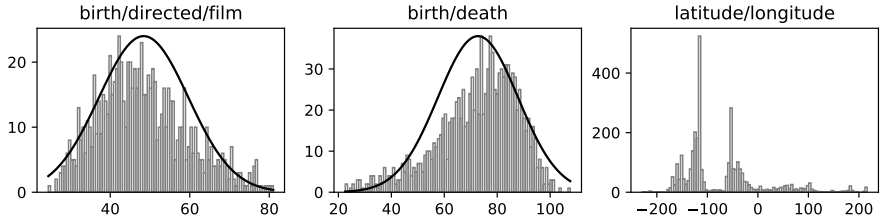


Figure 4.5 – Histograms and fitted normal curves of node attribute differences computed along some relations

For the message types between a pair of attributes of the same type and the ones expressed in same numerical range and unit *e.g.*, date attributes, the default value of parameter η is 1[III]. With this in mind, we plot the histograms of numerical attribute differences over some representative relationships in Figure 4.5. In the first two plots, we observe that the difference between `date_of_birth` of a person and `date_release` of the film directed by that person easily fits a normal distribution as well as the difference between `date_of_birth` and `date_of_death` of a person. Here, the mean corresponds to the estimated value of the parameter τ. The relation between these attributes empirically conforms the assumed local generative model in (4.1). On the other hand, `latitude` and `longitude` of a location do not accommodate such a correlation. Thus, MRAP can simply skip the message passing between such attributes. Given the number of attributes and relation types in each dataset, the total number of regression models actively used by MRAP is reported in Table 4.2. Given also the

[III] Since no scale change acts in the information exchange between such attributes, the parameter η in their message passing function is left by default.

number of multi-relational edges, it is possible to compute the number of message passing paths, which relates to the number of messages propagated across the graph in one iteration.

Table 4.2 – (Upper) Dataset statistics. (Lower) Characteristics of MRAP in these datasets.

	FB15K-237	YAGO15K
Entities	10,054	15,077
Edges	118,747	119,590
Relation types	114	32
Attribute types	11	7
Attributes in train	9,261	9,405
Attributes in validation	2,315	2,351
Attributes in test	2,315	2,351
Message passing paths	180,688	168,915
Message functions	310	261

Baselines. We compare MRAP to baseline methods introduced in [93]: GLOBAL and LOCAL. For each type of attribute, while GLOBAL replaces the missing values by the average of the known ones (mean imputation), LOCAL replaces them by the average of the known ones in the neighboring nodes. We also compare to NAP++ [93]. For each type of attribute, NAP++ constructs a k-NN graph upon the learned node embedding solely for the propagation of that type of attribute. As opposed to these methods, MRAP leverages the correlations across all attribute types and the multi-relational structure of the KG to impute the missing values.

Experimental Setup. Given KG datasets, we randomly split their node attributes into training, validation and test sets in a proportion of 80/10/10%. The validation set is used for the hyper-parameter tuning of NAP++ framework and we measure the performance of all methods on the test set. Statistics for this configuration are summarized in Table 4.2. We run experiments on several setups with different sparsity of observed node attributes. For this purpose, we use randomly subsampled versions of the training set as observed attributes and we set the rest as missing. To investigate the performance of MRAP, we report the results for two different setups: in the former, we use all of the training set as observed attributes and in the latter, we target a higher regime of sparsity and we use half of the training set as observed attributes. Throughout the section, we refer to these setups as '100%' and '50%' respectively.

Analysis. The performances of the baseline methods and MRAP on the two KG datasets are given in Table 4.3 and 4.4. First, it is worth to notice that the comparison of the methods across different setups (100% and 50%) is quite consistent. MRAP achieves competitive results against the other methods, specifically on date type of attributes, it performs mostly the best in both of the two datasets. We argue that this is achieved because MRAP profits the message passing between different types of attributes, unlike the other methods, which do not permit a direct information exchange between them. This is found to be critical particularly among the date attributes: when the message passing between different types of attributes is deactivated in MRAP, the prediction error for most of the date attributes raises. We run

Table 4.3 – Performances on FB15K-237 with two different setup of observed node attribute sparsity

	100 %						50 %					
	LOCAL/GLOBAL		NAP++		MRAP		LOCAL/GLOBAL		NAP++		MRAP	
Attribute	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**
`date_of_birth`	20.6	54.2	22.1	**34.3**	**15.0**	38.6	24.0	69.4	27.2	40.0	**12.3**	**20.5**
`date_of_death`	37.2	68.4	52.3	85.2	**16.3**	**32.2.2**	36.8	54.7	79.3	95.7	**16.0**	**25.2**
`film_release`	11.5	15.5	9.9	14.7	**6.3**	**8.6**	11.8	15.2	9.3	12.8	**6.4**	**9.0**
`organization_founded`	*73.3	*121.0	59.3	98.0	**58.3**	**91.6**	*72.3	*121.4	65.0	114.6	**60.9**	**96.5**
`location_founded`	138.0	*259.8	149.9	277.0	**98.8**	**151.9**	111.7	176.4	165.4	291.7	**105.9**	**146.2**
`latitude`	3.3	10.3	11.8	18.9	**1.5**	**3.5**	5.2	11.9	11.5	18.7	**2.1**	**4.1**
`longitude`	6.2	16.3	54.7	71.8	**4.0**	**8.8**	22.4	38.4	51.7	66.9	**4.7**	**9.3**
`area`	*5.4e5	***5.4e5**	**4.4e5**	1.2e6	4.4e5	1.1e6	*4.0e5	***4.1e5**	**3.2e5**	2.2e6	5.7e5	1.5e6
`population`	*7.7e6	***1.8e7**	**7.5e6**	6.5e7	2.1e7	4.3e7	***5.0e6**	***1.8e7**	7.5e6	6.4e7	2.3e7	4.2e7
`height`	*0.085	*0.104	**0.080**	**0.102**	0.086	0.106	*0.085	*0.104	**0.080**	**0.102**	0.087	0.108
`weight`	*14.2	*20.2	15.3	18.9	**12.9**	**18.3**	*14.2	*20.2	13.6	**17.3**	**13.2**	19.3

Table 4.4 – Performances on YAGO15K with two different setup of observed node attribute sparsity

	100 %						50 %					
	LOCAL/GLOBAL		NAP++		MRAP		LOCAL/GLOBAL		NAP++		MRAP	
Attribute	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**
`date_of_birth`	26.3	64.8	23.2	59.9	**19.7**	**31.5**	26.2	65.2	24.2	**61.3**	**21.1**	61.9
`date_of_death`	*48.6	*89.5	45.7	99.4	**34.0**	**84.2**	*45.4	*89.1	47.4	97.8	**35.0**	**84.4**
`date_created`	*95.5	*155.8	83.5	152.3	**70.4**	**149.6**	*96.0	*155.8	82.6	152.6	**65.8**	**135.3**
`date_destroyed`	42.2	**59.5**	38.2	75.5	**34.6**	62.0	41.8	59.3	33.9	68.3	**28.1**	**45.9**
`date_happened`	***52.1**	***67.3**	73.7	159.9	54.1	73.8	*60.1	***72.7**	77.0	141.5	**54.0**	95.6
`latitude`	3.4	9.0	8.7	13.8	**2.8**	**7.9**	6.7	14.7	9.2	14.2	**3.7**	**8.6**
`longitude`	10.6	24.1	43.1	58.6	**5.7**	**17.1**	20.5	34.6	45.2	60.9	**7.4**	**18.0**

additional experiments to justify other design choices of MRAP, and provide an ablation study in Table 4.5. First, we refer to the case where the message passing between different types of attributes is deactivated as 'w/o Cross' since this case blocks the information crossing from one attribute type to another. Second, we block the propagation of messages within a node, achieved by the inner loss term introduced in (4.4), and we refer to this case as 'w/o Inner'. Note that the former case, 'w/o Cross', already spans the latter, 'w/o Inner', because the inner-node message passing is always realized between different types of attributes. The experiments show that the cross-attribute and inner-node message passing enhances the prediction results almost always. We see that the inner-node message passing is significant in particular between the attributes `date_of_birth` and `date_of_death`, `area` and `population`, and then, `height` and `weight`. For instance, in the case 'w/o Inner', the error for the attribute `date_of_death` raises more than 10% as seen in Table 4.5.

In Table 4.3 and 4.4, LOCAL/GLOBAL reports the best performance obtained by either of the two baselines for each attribute and an asterisk (*) indicates that GLOBAL outperforms LOCAL. We see that GLOBAL performs the best for some types of attributes, *e.g.*, area and population. For the prediction of those, we argue that the underlying relational structure may

Table 4.5 – Ablation study for MRAP. MAE measured on the experimental setup '50%'.

Dataset	Attribute	w/o Cross	w/o Inner	MRAP
FB15K-237	date_of_birth	19.1	14.4	**12.3**
	date_of_death	41.0	20.0	**16.0**
	film_release	11.5	6.4	**6.4**
	organization_founded	71.0	**60.5**	60.9
	location_founded	148.7	106.1	**105.9**
	latitude	2.1	2.1	**2.1**
	longitude	4.7	4.7	**4.7**
	area	1.8e6	1.8e6	**5.7e5**
	population	2.4e7	2.4e7	**2.3e7**
	height	0.089	0.089	**0.087**
	weight	16.6	16.6	**13.2**
YAGO15K	date_of_birth	28.7	22.8	**21.1**
	date_of_death	52.4	42.7	**35.0**
	date_created	86.8	65.9	**65.8**
	date_destroyed	43.3	30.4	**28.1**
	date_happened	60.1	54.2	**54.0**
	latitude	3.7	3.7	**3.7**
	longitude	7.4	7.4	**7.4**

not be very informative, since the relation based methods, *i.e.*, LOCAL, NAP++, MRAP, perform poorly. The attributes with least number of samples (see Table 4.1) may also challenge the model parameter learning in NAP++ and MRAP and affect their performance. In addition, GLOBAL outperforms LOCAL occasionally, *e.g.*, `date_organization_founded` in FB15K-237 and `date_created` in YAGO15K. Even if the relational structure underlying those attributes are informative, LOCAL applies the neighborhood averaging regardless of the relation types. Here, MRAP improves the prediction by inducing heterogeneous local generative model.

Besides a better overall performance, MRAP exhibits other advantages with respect to NAP++: while MRAP performs the estimation of its parameters and the imputation of the missing values in seconds, NAP++ requires several hours, mostly due to the learning of node embeddings. The experiments are executed in a GTX Titan GPU. MRAP is also more efficient in memory—it only has to learn three parameters per regression function—as compared to NAP++, which learns a latent representation (whose dimensionality is 100) per node.

4.4.2 Performance of Semi-Supervised Learning Scheme

In this part, we provide comparison of 4 different frameworks for learning the numerical attributes in KGs. First, we test an instance of MRAP where we leave the parameters of the algorithm by default as $\tau = 0, \eta = 1, \omega = 1$ for each type of message. This obviously drops down to a standard label propagation algorithm for node attribute regression, which realizes message passing across all node attributes regardless of the heterogeneity of the underlying graph or the node attributes. Thus, we denote this framework simply as LP. Second, we test the

previously proposed MRAP framework, which requires the propagation parameters to be estimated in advance over the observations. Third, we test the proposed semi-supervised learning framework where we learn the parameters of the message passing functions—τ and η, however, we keep the weight parameter ω same for each type of message, hence it is deactivated. Throughout the discussions we refer to this framework as SSL-1. Last, we denote another framework as SSL-2 where we learn both the parameters of message passing functions and their corresponding weights simultaneously.

We compare the performances of the methods by the experiments conducted on a subset of YAGO15K dataset which composes its 5 different types of date attributes, *i.e.*, $|\mathcal{A}| = 5$. In this subset, we select 24 different types of relations, *i.e.*, $|\mathcal{P}| = 24$, which leads to 77 different types of messages—including inner node messages—each of which has at least 100 message passing paths composing one connected computational graph. The total number of message passing paths is 149607 whereas the total number of node attributes is 17488.

Experimental Setup. We again set the damping factor as $\xi = 0.5$ for all methods. On the other hand, we run a fixed number of iterations, which is set to 5 for each propagation framework. One should beware that it is above the diameter of the computational graph of message passing to ensure that there is no unpropagated node attribute left at the end of the forward pass.

We conduct the experiments by randomly setting 50% of the node attributes as known/observed, which can be considered as training attributes. Then, we measure the evaluation metrics over test attributes on 20 different instances of this setup and report the average of them for each method. Moreover, the aforementioned attention mechanisms are not used in SSL-2 since on the dataset we work the reparameterization asserted by the attention mechanisms goes beyond the number of the message types. Therefore, they do not bring further improvement on the performance of SSL-2. However, the proposed attention mechanism can be useful in training on higher level of heterogeneous datasets, where reparameterization drops down the number of message passing parameters.

Nonetheless, we confront an overfitting issue in training of SSL-2. To address that, we develop the following dropout technique.

Dropout Strategy. In training of deep neural nets, dropout is a common technique that is used by randomly deactivating some of the neurons in order to overcome overfitting to the training set and to provide generalization. By randomizing the dependency of the loss on the parameter set, this may destabilize the training procedure, however, it prevents the co-adaptation of the parameters. In graph convolution networks, such a strategy has been adapted as a message passing reducer [105]. We employ dropout in SSL-2 by randomly zeroing out the exchanged messages throughout the iterations in forward propagation. In particular, we select randomly 50% of the collected messages and exclude them joining the aggregations at each iteration except the last one. Dropout is only applied during training time while learning the parameters, and not used in test time.

For a sample training procedure, we plot the MSE loss over the training and test samples in Figure 4.6. The first pane shows the learning curve of SSL-1, where we train only the message functions. Here, we see a constant decrease in the loss over the training and test set and convergence as expected. On the other hand, we observe a different learning curve in training of SSL-2 in the middle pane, where we learn both the message passing functions and their weights. Here, the error on the training set keeps decreasing, while the error on the test set starts increasing after certain number of epochs. This signifies overfitting to the training set. Moreover, once we apply the dropout strategy in SSL-2, shown in the right pane, we obtain a more unstable learning curve, yet overall we gain a decrease in both training and test set.

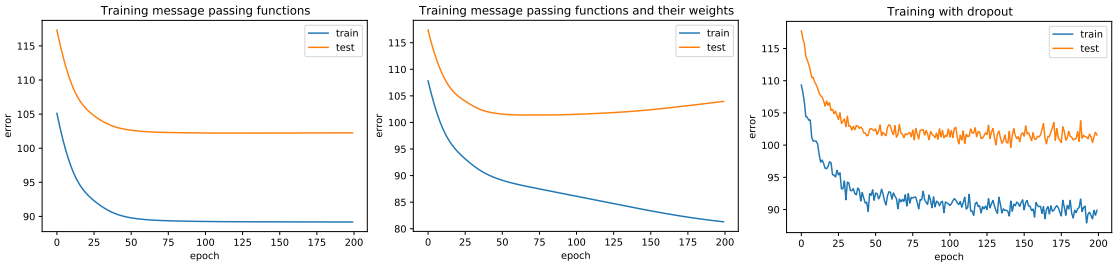


Figure 4.6 – Learning curves of SSL-1, SSL-2 without dropout, SSL-2 with dropout from left to right.

Table 4.6 – Performances of different learning frameworks on YAGO15K date attributes

	LP		MRAP		SSL-1		SSL-2	
Attribute	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**
`date_of_birth`	46.3	64.8	20.7	49.0	23.1	47.1	**20.0**	**44.4**
`date_of_death`	87.1	123.8	**42.0**	104.2	47.0	103.3	45.4	**99.1**
`date_created`	85.6	152.1	**65.4**	136.6	67.0	**134.3**	65.8	134.4
`date_destroyed`	94.8	130.7	50.4	**107.3**	49.2	111.9	**48.7**	113.0
`date_happened`	82.8	144.6	66.1	143.6	57.4	140.2	**52.6**	**136.1**
Overall	67.5	115.1	41.5	100.6	43.5	98.9	**41.2**	**97.8**

Analysis. In Table 4.6, we report the performance of SSL-2 by applying the dropout strategy. Here, we see that all the multi-relational frameworks outperforms the method LP by a substantial margin. Thus, we can claim that the heterogeneity of the data constitutes an important structural prior that one should exploit as an inductive bias in the node attribute completion task. In addition, semi-supervised learning frameworks in general manage to lower the prediction error beyond the method MRAP. Thus, we argue that learning the propagation parameters through back propagation seems a promising approach compared to estimating them over the observed pair of attributes. Overall, we achieve the best performance with SSL-2 where we learn the message passing functions and weights together by adopting the dropout strategy.

4.5 Conclusion

In this chapter, we address a relatively unexplored problem in knowledge graphs: node attribute completion. We present two alternative methods: a multi-relational propagation algorithm, MRAP, and a semi-supervised learning framework in order to complete missing numerical node attributes. The proposed propagation methods are framed in an heterogeneous message passing scheme, enabling information exchange across multiple types of attributes and over multiple types of relations. We show that MRAP very often outperforms several baselines in two datasets, whereas the preliminary results obtained by the proposed semi-supervised learning method assert that it can be a favorable alternative in certain data conditions. As a future work, we aim at broadening the experimental analysis of the proposed semi-supervised learning scheme by testing the effectiveness of the proposed compositional attention mechanisms in more challenging heterogeneity conditions.

In this work, we specifically study the regression of heterogeneous numerical features in a KG that are expressed in continuous values. Nonetheless, generalization of the proposed approaches integrating the categorical features and addressing the classification task also motivate future research directions.

The convenience of the proposed methods originates from the fact that the linear message functions and their weights render an interpretable and computationally simple learning scheme. More complex message functions are also possible if there are higher order dependency between the node attributes. For instance, the message functions can be designed as multi-layer perceptrons incorporating non-linear activation functions. This would yield a heterogeneous message passing neural net at the expense of the simplicity of the model, which constitutes the focus of our future work.

5 Graph Learning in Multi-Relational Data Domain

Structure inference is an important task for network data processing and analysis in data science. In recent years, quite a few approaches have been developed to learn the graph structure underlying a set of observations captured in a data space, we refer the reader to Chapter 2 Section 2.2 for a revisit. Although real-world data is often acquired in settings where relationships are influenced by a priori known rules, such domain knowledge is still not well exploited in structure inference problems. In this chapter, we identify the structure of signals defined in a data space whose inner relationships are encoded by multi-layer graphs. We aim at properly exploiting the information originating from each layer to infer the global structure underlying the signals. We thus present a novel method for combining the multiple graphs into a global graph using mask matrices, which are estimated through an optimization problem that accommodates the multi-layer graph information and a signal representation model. The proposed mask combination method also estimates the contribution of each graph layer in the structure of signals. The experiments conducted both on synthetic and real-world data suggest that integrating the multi-layer graph representation of the data in the structure inference framework enhances the learning procedure considerably by adapting to the quality and the quantity of the input data.

This chapter is organized as follows. We first give the motivation for our novel structure inference method, learning mask combination of multi-layer graphs, in comparison to the related learning schemes. Then, we introduce the settings that we work and the notation used throughout the chapter. We present our problem formulation and discuss it in detail. Finally, we give the experimental results and conclude. This chapter is based on a joint work with Dorina Thanou, Elif Vural and Pascal Frossard, titled: “Mask combination of multi-layer graphs for global structure inference" [38].

5.1 Mask Combination of Multi-layer Graphs

Many real-world data can be represented with multiple forms of relations between data samples. Multi-layer graphs are convenient for encoding complex relationships of multiple

types between data samples [18]. While they can be directly tailored from a multi-relational network such as a social network data, multi-layer graphs can also be constructed from a multi-view data [106, 107], where each layer is based on one type of feature.

In this study, we consider data described by a multi-layer graph representation where each data entity corresponds to a node on the graph along with signal values acquired on each graph node. Each graph layer accommodates a specific type of relationship between the data entities. From a multi-view data analysis perspective, we assume that the observed signals reside on a global view, which is latent, while the information about every single view is known. Ultimately, we aim at inferring the hidden *global graph* that best represents the structure of the observed signals.

Here, the task is to employ the partial information given by the multi-layer graphs to estimate the global structure of the data. For such a task, the connections contained in every layer may not have the same level of importance or multiple layers might have redundancy due to a correlation between them. Hence, it may cause information loss to consider a single layer as it is, or to merge all the layers at once [108]. In such cases, exploiting properly the information originating from each layer and combining them based on the targeted task may improve the performance of the data analysis framework.

Considering these challenges, we propose a novel technique to combine the graph layers, which has the flexibility of selecting the connections relevant to the task and dismissing the irrelevant ones from each layer. For this purpose, we employ a set of mask matrices, each corresponding to a graph layer. Through the mask combination of the layers, we learn the global structure underlying the given set of signals. The mask matrices are indicative of the contribution of each layer on the global structure. The problem of learning the unknown global graph boils down to learning the mask matrices, which is solved via an optimization problem that takes into account both the multi-layer graph representation and a signal representation model. The signal representation model typically depends on the assumption that the signals are smooth on the unknown global graph structure. A smooth signal generative model on graphs is introduced by Dong et al. [76]—please see Section 2.2.2 for a revisit, which we also adopt in our structure inference framework in multi-layer settings.

Fig. 5.1 illustrates the general framework with inputs which are signals captured on a set of data entities and a multi-layer graph representation storing the relations between those. The set of mask matrices, which forms the mask combination of graph layers, is an output together with a corrective term bridging the gap between the multi-layer graph representation and the signal representation model. The mask combination and the corrective term are summed up to yield the global graph. The ultimate output is the global graph best fitting the signals.

We run experiments on a multi-relational social network dataset and a meteorological dataset. In the proposed framework, the introduced set of observations determines how to combine the multi-layer graphs into a global graph. In the experiments on the meteorological data, for instance, we employ different types of measurements. When the type of the measurement

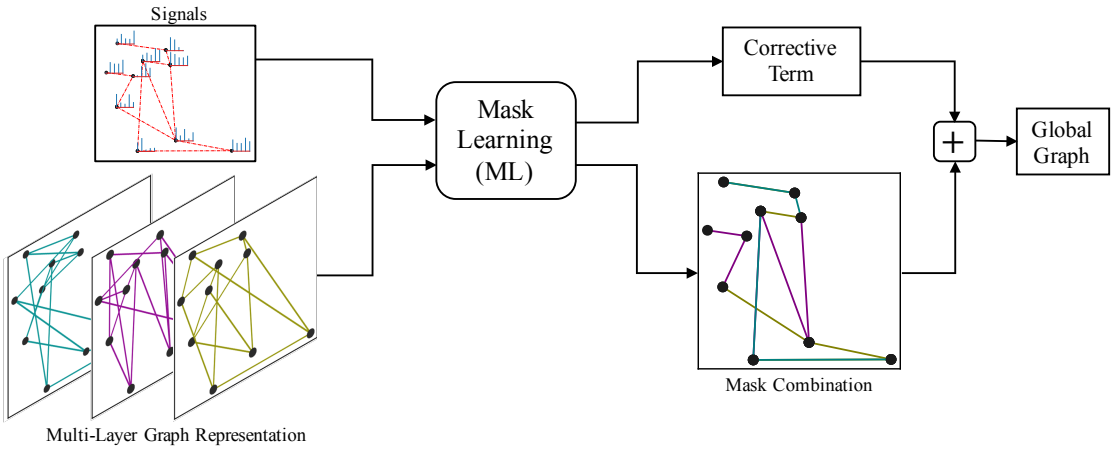


Figure 5.1 – An illustration for the input and output of the mask learning algorithm

is “temperature", the task is to infer the global structure underlying the temperature signals. Yet on the same set of weather stations, when “snow-fall" measurements are introduced as observations, then the task is to infer the global structure underlying the snow-fall signals, which is found to be different from that of temperature. Hence, the layer combination properly adapts to the target task and the inferred mask matrices uncover the relative importance of the layers in terms of structuring the signals of interest. In addition, our extensive simulation results suggest that, in a structure inference problem, exploiting the additional information given by the data domain through a multi-layer graph representation enhances the learning procedure by increasing its adaptability to variable input data quality.

5.1.1 Comparison to the Related Learning Schemes

In this section, we present a conceptual comparison of the proposed framework to the related ones i) studying combination of multiple graphs to accomplish network analysis or semi-supervised learning tasks, ii) adopting a graph regularization framework on multi-view data for semi-supervised learning or clustering tasks, iii) constituting the state-of-the-art structure inference schemes.

In the last decade, many studies have adopted multi-layer networks to treat the data emerging in complex systems ranging from biological networks to social networks, which promoted fundamental network analysis tools. In social networks, for instance, each type of relationship between individuals may be represented by a single layer and a specific combination of the layers may reveal hidden motifs in the network. For this purpose, Magnani et al. [108] propose the concept of power-sociomatrix, which adopts all possible combinations of the layers in the analysis of a social network. Considering multiple graph representations of a data space has also gained importance in some machine learning frameworks as well. For example, Argyriou et al. [109] propose adopting convex combination of Laplacians of multiple graphs encoding a data space for a subsequent semi-supervised learning task. For the same purpose, there

have been also several other works studying arithmetic mean [110] and generalized matrix means [111] of multiple graphs. From the topological perspective, such kind of combinations of multiple graphs yield identical set of solutions given by power-sociomatrix [108] since they treat a single graph as a whole, either keeping all its edges in the combination or dismissing. Thus, the layer combination do not have much flexibility in the topology. In our framework, on the other hand, the masking technique has the flexibility of selecting a particular set of edges from a layer to incorporate it in the layer combination.

Moreover, many studies have employed multiple graphs in order to represent the data emerging in multi-view domains and adapted the graph regularization framework to the multi-view domain in search of a consensus of the views [112, 113, 106, 114, 115]. Since most of those studies target the semi-supervised learning or clustering tasks, a low-rank representation of the data, which is common across the views, is sufficient. Lately, the authors in [116] developed a Graph Neural Network scheme to conduct semi-supervised learning on data represented by multi-layer graphs, where they integrate the graph regularization approach to impose the smoothness of the label information at each graph layer. In contrary to these methods, the proposed method specifically addresses a structure inference task which is achieved by the estimation of a graph underlying a set of observations/signals living on a multi-view/multi-layer data domain.

More recently, several graph regularization approaches have been proposed to learn a global or consensus graph from multi-view data for clustering [117, 118] and semi-supervised learning [119, 120]. They employ multi-view data to obtain a unified graph structure. Particularly in [117, 118], the authors propose optimization problems where single view graph representations are extracted first and then they are fused into a unified graph. Unlike in these learning schemes, the set of observations in our settings does not belong to a specific view of the data but they are assumed to reside on an unknown global view that we aim at inferring. In this sense, the study in [120] works in similar settings to ours. For a node classification task, it adopts a Graph Convolutional Network scheme defined on the merged graph that is obtained by adapting the method proposed in [106]. In our case, we rather obtain the so-called global graph through a novel technique that combines the given graph layers by flexibly adapting to the structure implied by the observed signals.

The problem of learning a graph representation of the data has been addressed by various network topology inference methods. We refer the reader to Section 2.2 in Chapter 2, for a review. Unlike the previous solutions learning the graph structure directly from a set of observations [74, 75, 76, 78, 77, 121], in our study we assume that multiple graphs representing the interactions between nodes at different levels are available, and we explicitly make use of this information while learning the global graph structure. The main benefit of the proposed method over those is that it can compensate for the often encountered case where we have a limited number of observations deviating from the assumed statistical model. Incorporation of the side information obtained from the multi-layer graph representation leads to a more reliable solution in such cases.

Although there are few graph learning algorithms [121, 122] allowing the incorporation of prior knowledge on the connectivity, the multi-layer domain information has not been exploited systematically in the existing structure inference approaches. Instead, there is a line a works [123, 124, 125, 126] addressing the inference of multiple graphs defined on a common node set from a collection of observation sets, each living on one graph. Unlike those, we aim at learning a single graph, the so-called global graph, with help of a priori known multi-layer graphs that encode the additional information given by the data domain. This brings certain advantages, especially when the signal representation quality is weak due to noisy data or insufficient number of observations, where a graph learning problem is relatively ill-posed. In addition to learning the graph structure of the signals, our framework infers the contribution of different layer representations of the data to the structure of the signals.

5.1.2 Contributions

This study proposes a novel structure inference framework that learns a graph structure from observations captured on a data domain with partial structural information. The main contributions are summarized as follows:

- The graph learning procedure is integrated with a multi-layer graph representation that encodes multi-relational information offered by the data domain.
- The task-relevant information is deduced effectively from each graph layer and combined into a global graph via a novel masking technique.
- The mask matrices are optimized on the basis of the task determined by the set of observations. Hence, they indicate the relative contribution of the layers.

5.2 Mask Learning Algorithm

We propose a structure inference framework for a set of observations captured on a node space, which can be represented by multi-layer graphs. We treat the observations captured on such a node space as *signals* whose underlying structure is described by the hidden *global graph*. Our task is to discover the global graph by exploiting the information provided by the multi-layer graph representation and the signals.

5.2.1 Multi-layer Graph Settings

Suppose that we have T graph layers, each of which stores a single type of relation between the data samples. We introduce a weighted and undirected graph to represent the relations on layer-t, $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t, \mathbf{W}_t)$ for $t \in \{1, 2, \cdots, T\}$, where $\mathcal{V}$ stands for the node set consisting of N nodes shared by all the layers, and, $\mathcal{E}_t$ and $\mathbf{W}_t$ indicate the edge set and the symmetric weight matrix for layer-t. A graph signal $\mathbf{x} \in \mathbb{R}^N$ can be considered as a function that assigns

a value to each node as $\mathbf{x}: \mathcal{V} \rightarrow \mathbb{R}$. We denote the set of signals defined on the node space $\mathcal{V}$ by a matrix $\mathbf{X} \in \mathbb{R}^{N \times K}$, which consists of K signal vectors on its columns. The signals in $\mathbf{X}$ are assumed to be smooth on the unknown global graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$. The Laplacian matrix of the global graph is further given by $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{W}$ is the global weight matrix. $\mathbf{D}$ is the corresponding degree matrix that can be computed as

$$\mathbf{D} = \text{diag}(\mathbf{W}\mathbf{1}),$$

where $\mathbf{1}$ is the column vector of ones and diag$(\cdot)$ forms a diagonal matrix from the input vector elements. $\mathbf{L}$ is a priori unknown but it belongs to the set of valid Laplacians, $\mathcal{L}$, that is composed of symmetric matrices with non-positive off-diagonal elements and zero row sum as

$$\mathcal{L} := \left\{ \mathbf{L} \in \mathbb{R}^{N \times N} \;\middle|\; \begin{array}{l} [\mathbf{L}]_{ij} = [\mathbf{L}]_{ji} \leq 0, \forall \{(i,j) : i \neq j\} \\ \mathbf{L}\mathbf{1} = \mathbf{0} \end{array} \right\}, \tag{5.1}$$

where $\mathbf{0}$ is the column vector of zeros.

5.2.2 Mask Combination of Layers

Adopting the multi-layer graph and signal representation model mentioned above, we cast the problem of learning the global graph as learning a combination of the graph layers. While each graph layer encodes a different type of relationship existing on the node space, the multiple graph layers might have some connections that are redundant or even irrelevant to the global graph structure. This requires occasional addition or removal of some edges from the layers while combining them into the global graph. For this purpose, we propose a masking technique, which has the flexibility to integrate the relevant information from layer topologies and to simultaneously adapt the global graph to the structure of the signals. We introduce the combination of layers as a masked sum of the weight matrices of the graph layers:

$$\mathbf{W}_M = \sum_{t=1}^{T} \mathbf{M}_t \odot \mathbf{W}_t, \tag{5.2}$$

where $\odot$ represents the Hadamard (element-wise) product between two matrices: the weight matrix of $\mathcal{G}_t$, which is denoted as $\mathbf{W}_t$, and the symmetric and non-negative mask matrix $\mathbf{M}_t$ associated with layer $\mathcal{G}_t$. The mask matrices are stacked into a variable as $\mathbf{M} = [\mathbf{M}_1 \cdots \mathbf{M}_T]$, which is eventually optimized to infer the global graph structure. In general, the relations given in different layers may not have the same importance in the global graph. Hence, for an edge between node-i and node-j, the proposed algorithm learns distinct mask elements at each layer, for instance $[\mathbf{M}_t]_{ij}$ at layer $\mathcal{G}_t$ and $[\mathbf{M}_u]_{ij}$ at layer $\mathcal{G}_u$.

We finally define a function $\Lambda(\mathbf{M})$ to compute the Laplacian matrix of the mask combination

given by a set of mask matrices $\mathbf{M}$ as follows:

$$\Lambda(\mathbf{M}) = \text{diag}(\mathbf{W}_M \mathbf{1}) - \mathbf{W}_M. \tag{5.3}$$

5.2.3 Problem Formulation

Our task now is to infer the global graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$, on which the signal set $\mathbf{X}$ has smooth variations. Hence, in the objective function, we employ the well-known graph regularizer term $\text{tr}(\mathbf{X}^\intercal \mathbf{L}\mathbf{X})$, which measures the smoothness of the signal set $\mathbf{X}$ on the global graph Laplacian $\mathbf{L}$—we refer the reader to Section 2.1.1 to revisit of the origin of the graph regularization term. The optimization problem boils down to learning a set of mask matrices, $\mathbf{M}$. Within certain masking constraints, it captures the connections that are consistent with the structure of the signals from the multi-layer graph representation and yields a mask combination of the layers. In addition, we introduce a corrective term, $\mathbf{L}_E$, which makes a transition from the mask combination obtained from the given layers to the global graph that fits the observed signals within the smooth signal representation model. By summing it with the Laplacian of the mask combination, we express the global graph Laplacian as

$$\mathbf{L} = \Lambda(\mathbf{M}) + \mathbf{L}_E,$$

which is the ultimate output of the algorithm. The Frobenius norm $\|\cdot\|_F$ of $\mathbf{L}_E$ permits to adjust the impact of the corrective Laplacian, $\mathbf{L}_E$, on the global graph. The overall optimization problem is finally expressed as follows:

$$\begin{aligned}
\min_{[\mathbf{M},\mathbf{L}_E]} \quad & \text{tr}(\mathbf{X}^\intercal \Big(\Lambda(\mathbf{M}) + \mathbf{L}_E \Big) \mathbf{X}) + \gamma \|\mathbf{L}_E\|_F^2 \\
\text{s. t.} \quad & [\mathbf{M}_t]_{ij} = [\mathbf{M}_t]_{ji} \geq 0, t = \{1, 2, \cdots, T\}, \forall (i, j) \\
& \sum_{t=1}^{T} [\mathbf{M}_t]_{ij} = 1, \forall (i, j) \\
& \Lambda(\mathbf{M}) + \mathbf{L}_E \in \mathcal{L} \\
& \text{tr}(\Lambda(\mathbf{M}) + \mathbf{L}_E) = \Gamma,
\end{aligned} \tag{5.4}$$

where γ is a hyperparameter adjusting the contribution of $\mathbf{L}_E$ on $\mathbf{L}$. The last constraint on $\text{tr}(\Lambda(\mathbf{M}) + \mathbf{L}_E)$, the trace of the global graph Laplacian $\mathbf{L}$, fixes the volume of the global graph. It is set to be a non-zero value, i.e., $\Gamma > 0$, in order to avoid the trivial solution, i.e., null global graph. It can be considered as the normalization factor fixing the sum of all the edge weights in the global graph so that the relative importance of the edges can be interpreted properly. The mask matrices are then constrained to be symmetric and non-negative, which leads to a symmetric mask combination, $\Lambda(\mathbf{M})$. The global graph Laplacian, $\mathbf{L}$, is constrained to be a valid Laplacian. Consequently, $\mathbf{L}_E$ is forced to be a symmetric matrix but it does not have to be a valid graph Laplacian matrix. In this regard, $\mathbf{L}_E$ provides the possibility to make a subtraction from the mask combination as well as to add more weights on top of the mask combination. We also put a constraint on the mask elements $\{[\mathbf{M}_t]_{ij}\}_{t=1}^{T}$, which sets a search

space of the mask matrices yielding unit sum. This establishes a dependency between the mask elements corresponding to the same edge at each layer so that the contribution of the layers at a particular connection between node-i and node-j is normalized. As a result of the unit sum constraint on masks, the weight elements of the mask combination, given in (5.2), are confined into the weight range delivered by the layers as follows,

$$\min_t [\mathbf{W}_t]_{ij} \leq [\mathbf{W}_M]_{ij} \leq \max_t [\mathbf{W}_t]_{ij}. \tag{5.5}$$

Such a restriction is actually important to keep the weight values of the global graph in a reasonable range, which is desired for the weight prediction task. Note that dismissing an arbitrary edge $\mathcal{E}_{ij}$ from the mask combination is possible if

$$\min_t [\mathbf{W}_t]_{ij} = 0,$$

i.e., a connection is not defined between node-i and node-j in at least one of the layers. Also due to the unit sum constraint on the mask coefficients, the edge set of the mask combination is confined to the intersection and the union of the layer edges. In other words, the intersecting edges across the layers are kept in the mask combination—also apparent in (5.5)—but not necessarily in the global graph due to the corrective term.

The objective function in (5.4) is linear with respect to the mask matrices $\mathbf{M}$ due to the first term, and it is quadratic with respect to the corrective Laplacian $\mathbf{L}_E$ due to the second term. All the constraints are linear with respect to the optimization variables. Therefore, the problem is convex and it can be efficiently solved by quadratic programming.

5.2.4 Discussion

A theoretical analysis of the proposed problem is presented in this section, regarding the selection of the hyperparameters, the complexity and the identifiability.

Hyperparameters

In problem (5.4), we need to set two hyperparameters: γ and Γ. First, γ adjusts the impact of the corrective Laplacian, $\mathbf{L}_E$, on the global graph Laplacian, $\mathbf{L}$. As γ approaches infinity, there is a full penalty on $\mathbf{L}_E$, hence the problem (5.4) behaves as a constrained optimization problem where $\mathbf{L}_E$ is null, i.e., $\mathbf{L}_E = \mathbf{O}$. In the other extreme case where $\gamma = 0$, the global graph structure is completely defined by $\mathbf{L}_E$, which cancels out all the edges on the mask combination, $\Lambda(\mathbf{M})$, and leaves only a few edges constituting the links along which the signals are the smoothest. In this regard, γ should be set strictly above 0 in order to exploit the multi-layer graph representation adequately. The hyperparameter γ is used for the purpose of interpolating the solution between the support of the multi-layer graph representation and the agreement of the signal representation. As depicted on the limit cases, the maximum exploitation of the multi-layer graph representation can be obtained when γ approaches to

infinity where the corrective term has no contribution and the global graph is directly equal to the mask combination. Broadly speaking, γ should be set to a high value, when the input multi-layer graph representation is more reliable than the observations. Then, smaller values should be preferred when the observations are more informative so that the mask combination is refined by the corrective term according to the agreement of the signal representation. The factors playing a role in the quality of the input data also affect the accuracy of the proposed algorithm and they will be explained in the next part in detail. Also, note that the value of γ should be chosen proportionally to the squared norm of the observation matrix $\mathbf{X}$ due to the interplay between the first and the second term of the objective function in (5.4).

Second, the value of the parameter Γ sets the volume of the global graph. Recall that the masking constraints confine the edge weights of the mask combination into the interval given by edge weights of the layers, as stated in relation (5.5). Inherently, the volume of the mask combination, i.e., $\sum_{i,j}[\mathbf{W}_M]_{ij}$, is confined to the range given by the layer weight matrices. Γ can be considered as a budget on the volume of the edges to be masked from the given graph layers together with the volume of the corrective term. Accordingly, the number of edges in the global graph is proportional to the value of Γ as a consequence of the proposed masking approach. For the set of solutions where $\mathbf{L}_E = \mathbf{O}$, Γ is subject to the same feasible range for the volume of the mask combination $\mathbf{W}_M$. In that case, it has to be set as,

$$\sum_{i,j} \min_t [\mathbf{W}_t]_{ij} \leq \Gamma \leq \sum_{i,j} \max_t [\mathbf{W}_t]_{ij}, \tag{5.6}$$

so that $\mathbf{M}$ can be solved. The lower limit corresponds to the topology composed of the common edges across the layers and the upper limit corresponds to the topology given by the union of the layers. Recall that $\mathbf{L}_E$ is solved as a null matrix usually when γ in (5.4) is very large, which acknowledges the full reliability on the multi-layer graphs by pushing the global graph to have the topology and the weight range provided by the layers. Decreasing the value of γ relaxes this restriction, which enlarges the solution space for the global graph by diverting it from the mask combination solution. To conclude, Γ has a direct effect on the sparsity of the global graph. In practice, it can be chosen so as to ensure the desired sparsity level and in the feasible range of the volume of the mask combination determined by the layers as given in (5.6).

Complexity Analysis

The algorithm solves for the optimization variables consisting of the elements of the mask matrices $\{\mathbf{M}_t\}_{t=1}^{T}$ for T layers and the elements of the corrective Laplacian matrix, $\mathbf{L}_E$. The number of optimization variables for mask elements is $O(\sum_t |\mathcal{E}_t|)$, which is the sum of the number of edges given by the layers. It can also be written as $O(ET)$, where E is the average number of edges given by the layers. In the worst case, all the given layers are complete graphs where $E = \frac{N(N-1)}{2}$. However, typically, the given graph layers are sparse. If we assume that the average number of neighbors for a node in a graph layer is $k \ll N$, which makes $E = kN$, then we can say that the number of optimization variables for the mask elements

grows linearly as $O(kNT)$. Second, the corrective term, $\mathbf{L}_E$, has $\frac{N(N-1)}{2}$ elements. Thus, the objective function depends on $O(N^2)$ variables quadratically and $O(kNT)$ variables linearly, which makes $O(kNT+N^2)$ in total. The number of the optimization variables has a quadratic asymptotic growth with respect to the number of nodes, N. It is dominated by the elements of $\mathbf{L}_E$ when $kT < N$. Moreover, due to the fact that the objective function depends quadratically on $\mathbf{L}_E$, solving for these $O(N^2)$ variables also dominates the complexity, which implies that N is the factor of the complexity rather than T. The objective function in (5.4) is subject to a set of equality and inequality constraints expressed on the variables $\mathbf{M}$ and $\mathbf{L}_E$, which narrows down the solution space considerably. Ultimately, the overall complexity is determined by the quadratic programming, whose computational analysis for SDPT3 solver is given in [127].

In particular, one might desire to solve the problem in (5.4) in such a way that the global graph relies entirely on the multi-layer graph representation, where the corrective term, $\mathbf{L}_E$, has no contribution. This can be realized by choosing the hyperparameter γ above a certain large value. Furthermore, in certain applications, e.g., involving large networks, due to limitations on computational resources, one may also prefer to set $\mathbf{L}_E = \mathbf{O}$ and to be exempted of solving it completely. This is possible by using a reduced version of (5.4) where $\mathbf{M}$ is the only optimization variable, and it is expressed by the following optimization problem:

$$\begin{aligned} \min_{\mathbf{M}} \quad & \operatorname{tr}(\mathbf{X}^\intercal \Lambda(\mathbf{M})\mathbf{X}) \\ \text{s. t.} \quad & [\mathbf{M}_t]_{ij} = [\mathbf{M}_t]_{ji} \geq 0, t = \{1,2,\cdots,T\}, \forall (i,j) \\ & \sum_{t=1}^{T} [\mathbf{M}_t]_{ij} = 1, \forall (i,j) \\ & \operatorname{tr}(\Lambda(\mathbf{M})) = \Gamma, \end{aligned} \tag{5.7}$$

which can be solved via linear programming. The objective function of this problem is equivalent to that of (5.4) where $\mathbf{L}_E = \mathbf{O}$. Then, the equality and inequality constraints become equivalent to those of (5.4) when $\mathbf{L}_E = \mathbf{O}$, noting that the first constraint in (5.7) already implies $\Lambda(\mathbf{M}) \in \mathcal{L}$. Relying on these facts, we can say that uniting the solution space of (5.7) with $\{\mathbf{L}_E = \mathbf{O}\}$, we obtain a subset of the solution space of the problem (5.4). The reduced version requires only $O(kNT)$ optimization variables, which depends linearly on the number of nodes, N, hence, decreases the computational complexity considerably compared to the original problem. As a comparison, we finally note that, the optimization variables of the graph learning problems mentioned in Section 2.2 are subject to $O(N^2)$ in general.

Identifiability Analysis

The accuracy of the proposed learning scheme depends on the quality and the quantity of the input data. First of all, the factors playing a role in signal representation quality can be counted as follows:

- the ratio of the number of observations to the number of nodes (K/N). The accuracy

of the statistical inference built upon the smooth signal representation model is better when there are many observations in comparison to the data dimension.

- The signal-to-noise (SNR) of the signal set. The accuracy is better when there are clean signals that are sufficient to support the smooth signal representation model.
- The correlation between the observations. The accuracy is better when the observations are independent and identically distributed (i.i.d.).

Note that the accuracy of the graph learning methods mentioned in Section 2.2 are also subject to the facts above [34]. However, theoretical guarantees of the graph Laplacian estimation methods regarding the rate of convergence and error bounds are not well explored in terms of the listed factors. Nonetheless, the graph Laplacian can be counted as a specific instance of the precision matrix of the observations [34] and there have been several works [128, 129] studying the problem of estimating normal precision matrices in more general settings. Our algorithm estimates the graph Laplacian under quite particular priors based on a multi-layer graph structure, therefore, it is not straightforward to express a theoretical guarantee particularly fitting to our algorithm. Yet, we argue that the benefit of the proposed learning scheme over the aforementioned graph Laplacian inference algorithms is that it does not only depend on the observations but it also profits from the information originating from the multi-layer graph representation of the data. This is advantageous especially when the signal representation quality is not fully accountable. Accordingly, the accuracy of the proposed method depends on the multi-layer graph representation quality as well. Some related parameters are:

- the proportion of the global graph edges, $\mathcal{E}$, that are given by the layer edges $\mathcal{E}^L$, which can be measured by a term called *coverability* introduced in [108]. Coverability is the recall of the multi-layer graph representation on the global graph, and it is calculated by $\frac{|\mathcal{E} \cap \mathcal{E}^L|}{|\mathcal{E}|}$. It measures how much the multi-layer graph representation covers the global graph and it is 1 when the global graph is fully covered by the layers.
- the proportion of the common edges across the layers that are present in the global graph. This is due to the fact that the intersecting edges across the layers are present in the mask combination—but not necessarily in the global graph—by the mask constraints.

It is possible to relax the effect of these factors on the accuracy by choosing relatively small values of γ and fit the global graph more with respect to the information emerging from the observations.

5.3 Experiments

We compare the global graph recovery performance of our method (**ML**) against some state-of-the-art graph learning algorithms. First, we compare the graph learning algorithm that

we consider as baseline [76], which is referred to as **GL-SigRep**— please see the problem in (2.41) for a revisit. To make a fair assessment, we compare our method to another version of **GL-SigRep**, where the graph learning algorithm is informed of the input layers by restricting its solution space to the set of edges given by the layers as below:

$$\begin{aligned} \min_{\mathbf{L}} \quad & \operatorname{tr}(\mathbf{X}^\top \mathbf{L}\mathbf{X}) + \gamma \|\mathbf{L}\|_F \\ \text{s. t.} \quad & \mathbf{L} \in \mathcal{L} \\ & \operatorname{tr}(\mathbf{L}) = N \\ & [\mathbf{L}]_{ij} = 0, \text{ for } \{(i,j) : [\mathbf{W}_t]_{ij} = 0, \forall\, t\}. \end{aligned} \tag{5.8}$$

We refer to this method as **GL-informed**.

We also compare against the optimal convex combination of the layers. For that purpose, we adapt the method for learning the convex combination of multiple graph Laplacians introduced in [109] for our settings as in the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \operatorname{tr}(\mathbf{X}^\top \mathbf{L}\mathbf{X}) + \beta \|\boldsymbol{\alpha}\|_2^2 \\ \text{s. t.} \quad & \mathbf{L} = \sum_{t=1}^{T} \alpha_t \mathbf{L}_t \\ & \alpha_t \geq 0, \forall\, t \\ & \sum_{t=1}^{T} \alpha_t = 1, \end{aligned} \tag{5.9}$$

where the coefficients $\boldsymbol{\alpha} = [\alpha_1 \cdots \alpha_T]$ are learned for the convex combination of the layer Laplacians, $\{\mathbf{L}_t\}_{t=1}^{T}$, to reach the global graph Laplacian $\mathbf{L}$. Throughout this section, the algorithm solving the problem (5.9) is referred to as **GL-conv**.

For the quantitative assessment of link prediction performance, we employ the following evaluation metrics: *Precision, Recall* and *F-score* [130]. We also compute the *mean squared error (MSE)* of the inferred weight matrix for the assessment of weight prediction performance. We solve the problems **ML** (5.4), **GL-informed** (5.8), **GL-SigRep** [76] and **GL-conv** (5.9) via quadratic programming for which we utilize the CVX toolbox [131] with SDPT3 and MOSEK [132] solver and the code is available online[I].

5.3.1 Experiments on Synthetic Data

In this section, we run experiments on two different scenarios. First, we generate the global graph in a fully complementary scenario where the mask combination of the layers is directly equal to the global graph. Second, we test the algorithms on a non-fully complementary scenario where the global graph is created from a perturbation on the topology of the mask combination. For both cases, we generate the mask combination and the signal set as follows:

[I] https://github.com/bayrameda/MaskLearning

Generation of layers and the mask combination. First, the node set $\mathcal{V}$ is established with $|\mathcal{V}| = N$ nodes whose coordinates are generated randomly on the 2D unit square with a uniform distribution. Next, an edge set $\mathcal{E}^L$ is constructed for the layers by putting edges between all pairs of nodes in $\mathcal{V}$ whose Euclidean distance is under a certain threshold. The edge weights are computed by applying a Gaussian kernel, i.e., $\exp(-d(i,j)^2/2\sigma^2)$, where $d(i,j)$ is the distance between node-i and node-j and $\sigma = 0.45$. To generate two graph layers, $\mathcal{V}$ is randomly separated into two neighborhood groups: $\mathcal{V}_1$ and $\mathcal{V}_2$. Let us denote the set of edges connecting the nodes in one group $\mathcal{V}_t$, to all nodes in $\mathcal{V}$ as $\mathcal{E}^L_{\mathcal{V}_t,\mathcal{V}}$. The graph layer $\mathcal{G}_t$ is built on the edge set $\mathcal{E}_t = \mathcal{E}^L_{\mathcal{V}_t,\mathcal{V}}$, and the corresponding edge weights are used to construct its weight matrix $\mathbf{W}_t$. For the generation of the masks, another set of edges $\mathcal{E}^M$, a subset of $\mathcal{E}^L$ whose edge weights are above $\tau = 0.8$, are reserved. Let us denote the set of edges in $\mathcal{E}^M$ that are between a pair of nodes in $\mathcal{V}_t$ as $\mathcal{E}^M_{\mathcal{V}_t,\mathcal{V}_t}$. The mask matrix $\mathbf{M}_t$ is constructed by setting its entries corresponding to the edges in $\mathcal{E}^M_{\mathcal{V}_t,\mathcal{V}_t}$ as 1. Also, all the entries corresponding to the common edges between the layers, $\mathcal{E}_1 \cap \mathcal{E}_2$, are set as 0.5 in the mask matrices in order to keep the intersection of the layers in the mask combination. Lastly, the weight matrix of the mask combination is computed via the formulation given in (5.2). As the next step, the global graph is produced according to one of the experimental scenarios that will be explained in the following sections.

Signal Generation. Following the generation of the mask combination and the global graph, the global graph Laplacian matrix, $\mathbf{L}$, is computed. Using that, a number of smooth signals are generated according to the generative model introduced in [76]—please also see Section 2.2.2 in Chapter 2 for a revisit. Basically, the graph Fourier coefficients $\mathbf{h}$ of a sample signal can be drawn from the following distribution;

$$\mathbf{h} \sim \mathcal{N}(0, \Sigma) \tag{5.10}$$

where Σ is the Moore-Penrose pseudo-inverse of $\Sigma^\dagger$, which is set as the diagonal eigenvalue matrix of $\mathbf{L}$. The eigenvalues, which are associated with the main frequencies of the graph, are sorted in the main diagonal of $\Sigma^\dagger$ in ascending order. Thus, the signal Fourier coefficients corresponding to the low-frequency components are selected from a normal distribution with a large variance while the variance of the coefficients decreases towards the high-frequency components. In other words, the signal is produced to have most of its energy in the low frequencies, which enforces smooth variations in the expected signal over the graph structure. A signal vector is then calculated from $\mathbf{h}$ through the inverse graph Fourier transform [28]—see Eq. (2.39).

Fully Complementary Scenario

We first conduct experiments where the global graph is directly equal to the mask combination. We refer to this data generation setting as the fully-complementary scenario since the edge set of the global graph is fully covered by the union of the layers, thus, the coverability is fixed to 1. We generate 50 smooth signals on the global graph. Its volume is normalized by the number

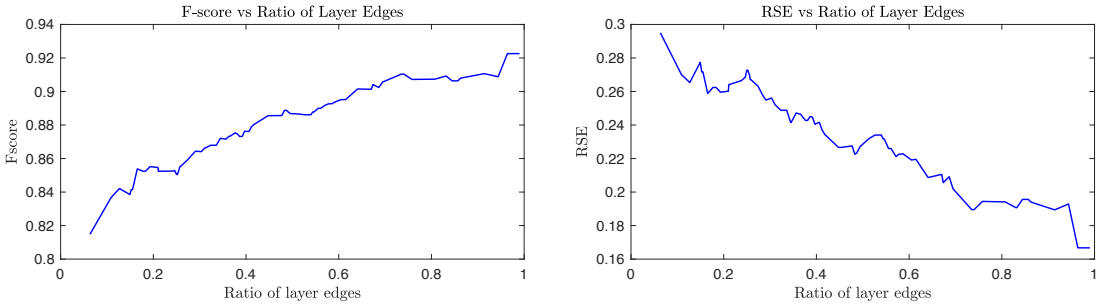


Figure 5.2 – Performance with respect to the ratio of layer edges

of nodes, $N = 20$. **GL-informed** (5.8) already learns a graph with a volume of N, therefore, we set the parameter $\Gamma = N$ in **ML** as well. The volume of the graph learned by **GL-conv** (5.9) is also normalized to N for a fair comparison of the MSE score. This experimental scenario—generating randomly the fully complementary layers, the global graph and the signal set in aforementioned settings—is repeated 20 times and the performance metrics are averaged on these 20 instances. The findings are summarized in Table 5.1. Following the discussion in Section 5.2.4, we employ the reduced version of **ML** in (5.7), since the corrective term $\mathbf{L}_E$ is not required in the fully-complementary settings. Consequently, the global graph is inferred to be directly equal to the mask combination. In Table 5.1, **GL-conv** yields a high difference between the recall and the precision rate since it either picks the edge set of a layer as a whole or not. Therefore, it is not able to realize an edge-specific selection, which leads to poor F-score compared to other methods. The global graph recovery performance of **GL-informed** is presented as a surrogate of **GL-SigRep**, since the solution for the global graph already lies in the edge set given by the layers in fully-complementary settings. The MSE score of **ML** and **GL-conv** is better than the one of **GL-informed**. This is due to the fact that **ML** and **GL-conv** have better guidance on the weight prediction task by confining the interval of weight values of the global graph to the interval introduced by the layers, which is expressed in (5.5) for **ML**. Finally, **ML** achieves good rates on the mask recovery performance, which measures how

Table 5.1 – Global Graph Recovery and Mask Recovery Performances

		precision	**recall**	**F-score**	**MSE**
Global Graph Recovery	**ML**	86.98%	90.79%	**88.84%**	**1.6E-03**
	GL-informed	81.26%	88.91%	84.48%	2.6E-03
	GL-conv	63.82%	100%	77.41%	2.1E-03
Mask Recovery	**ML**	92.57%	94.88%	93.68%	-

correctly the algorithm selects the edges from each layer to form the mask combination.

In this setting, in each repetition of the experiment, the number of edges given by each layer is also recorded to see the effect of the ratio of the layer edges e.g., $|\mathcal{E}_1|/|\mathcal{E}_2|$, on the performance of **ML**. Note that, $|\mathcal{E}_1|/|\mathcal{E}_2| = 1$ means that the layers are completely balanced and $|\mathcal{E}_1|/|\mathcal{E}_2| = 0$

means that one layer is completely deficient in terms of the number of edges. Here, we employ the *relative squared error* (RSE) as a metric to assess the change in the accuracy of weight matrix estimation, which is the normalized form of the squared error by the squared norm of the ground truth weight matrix. In Fig. 5.2, we see that the performance is enhanced approximately by 12% when the layers are balanced compared to the deficient layer case. We argue that the reason for such an enhancement is the improvement in the alignment between the layers, considering the fact that the masking coefficients are constrained in a way to keep the intersecting edges between the layers in the mask combination. In other words, when the layers are more balanced, there is a higher chance of a larger intersection. Hence, we speculate that balanced layers may lead to better performance as long as the alignment between the layers is important for the structure of the observations, as in the case of the synthetic data generated in fully complementary settings. In the extreme case where the intersection is empty when one layer is completely deficient, the performance obviously loses the gain that could be obtained from the overlap between the layers.

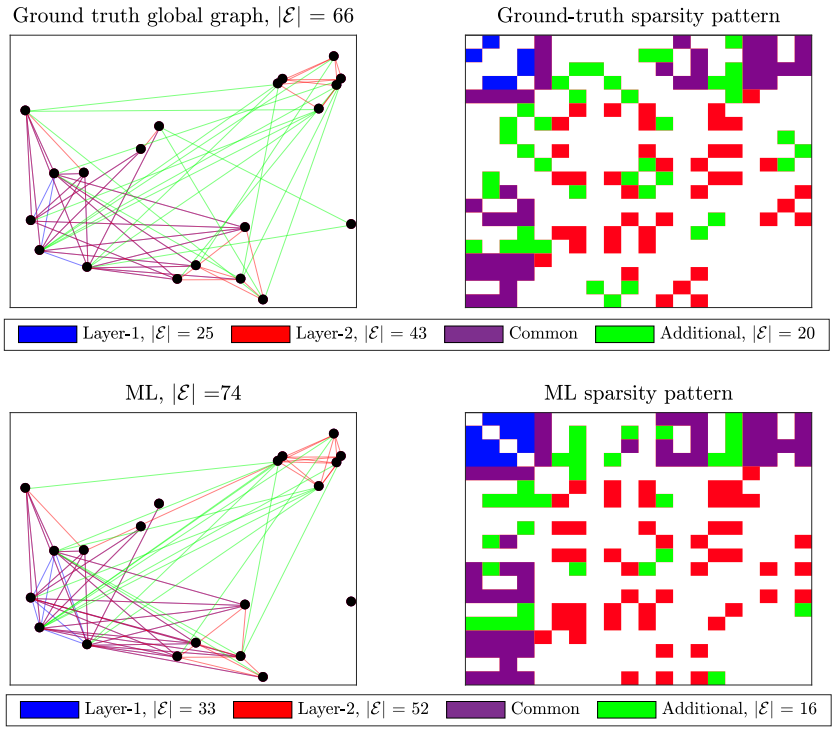


Figure 5.3 – Ground truth global graph and the solution given by **ML**

Non-fully complementary scenario

In this section, we test the algorithms in experiments where the data is generated with different levels of multi-layer and signal representation quality so that we analyze their effects on the global graph recovery performance. First, to create the global graph, we deviate from the exact mask combination by perturbing its topology to some degree. Basically, we randomly replace

a set of edges existing on the mask combination outside the union of the graph layers. The degree of such a perturbation on the mask combination can be measured by the coverability. The larger the number of edges perturbed on the topology of the mask combination, the more the global graph diverts from the multi-layer graph representation, which decreases the coverability. Consequently, the multi-layer representation quality drops. A demonstration is provided in Fig. 5.3 top row where the global graph is generated with coverability 0.7. Here, the set of edges outside the mask combination is shown in green. As seen in Fig. 5.3 bottom row, **ML** manages to predict some edges that are not given by the multi-layer graph representation, thanks to the corrective term in (5.4).

Effect of multi-layer representation quality. Here, we test the performance of **ML** in non-complementary settings with different coverability and different values of γ. We conduct each experiment with signal sets composed of 50 signals that are generated on the global graph as explained before. We average the performance metrics on 20 experiments in Fig. 5.4. The

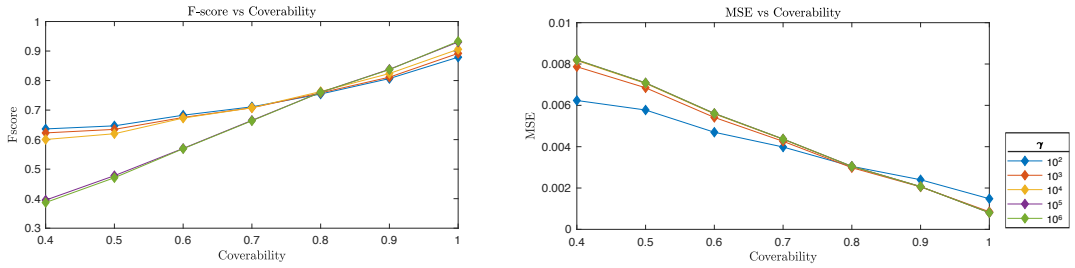


Figure 5.4 – Performance of **ML** with different γ values vs coverability

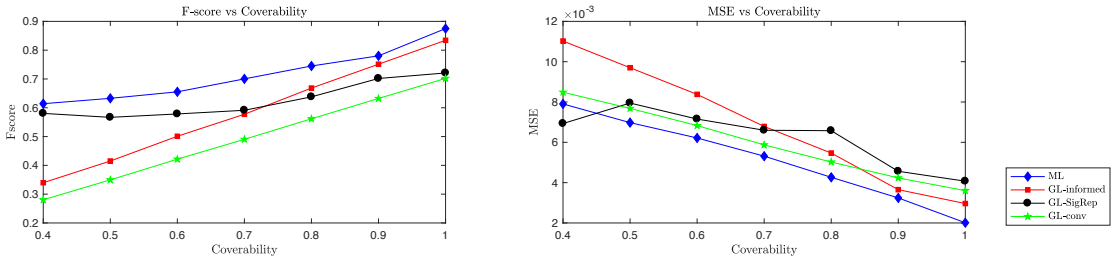


Figure 5.5 – Performance of the algorithms vs coverability

following observations can be made: (i) When coverability has the lowest value (0.4), **ML** with $\gamma = 100$ has the best performance. (ii) When it has the highest value (1), which corresponds to fully complementary settings, **ML** with $\gamma = 10^6$ has the best performance. (iii) Whatever value is chosen for the parameter γ, the performance of **ML** gets better with increasing coverability. Considering these facts, choosing a smaller value for the parameter γ seems to be a good remedy for lower coverability settings. Yet, this degrades the performance slightly in the high coverability settings, which confirms the theoretical analysis given in 5.2.4. Hence, if there is no prior knowledge on the reliability of the multi-layer graph representation or the signal representation, one may prefer to use small values for γ by compromising a small decay in

the performance in the case of highly reliable multi-layer graph representation. Moreover, the performance of **ML** improves as the global graph approaches the mask combination of the layers. This is simply because the algorithm bases the global graph on top of the mask combination, and any modification made on it by the corrective term is subject to an extra cost and thus limited. Therefore, **ML** with any γ value performs best when the mask combination is directly equal to the global graph, which is possible only in the fully complementary settings. Still, the corrective term improves the performance in the non-fully complementary settings. Given the plots in Fig. 5.4, an appropriate γ value for each coverability interval can further be found. For example, it can be chosen as $\gamma = 100$ for coverability ≤ 0.75, then $\gamma = 10^4$ until coverability $= 0.8$, $\gamma = 10^5$ later until coverability $= 0.9$ and $\gamma = 10^6$ for coverability > 0.9. We now adopt these values to present the performance of **ML** against the competitor algorithms by averaging the performance metrics on 20 experiments in each coverability setting, given in Fig. 5.5. Beginning with the performance of **GL-informed**, we see that its performance improves regularly with the raising coverability, and it outperforms **GL-SigRep** for coverability ≥ 0.73. The coverability is irrelevant for the performance of **GL-SigRep** since it receives no multi-layer guidance, hence the fluctuations can be disregarded as the coverability changes. Nonetheless, its performance slightly drops in low coverability settings. This is because the edges of the global graph are rewired randomly outside the union of the layers , which renders the graph towards a random network. It is acknowledged in [76] that graph learning from smooth signals in random network structures has slightly lower performance than learning on regular networks. Still, in Fig. 5.5, the performance of **GL-SigRep** in black line should be considered as a reference since it is the least affected by the coverability. Furthermore, the trend of **ML** in blue line seems to be more resistant than **GL-informed** in low coverability settings, thanks to the corrective term. The performance of **ML** approaches **GL-SigRep** as coverability decreases since the multi-layer guidance diminishes. Yet, it manages to keep its F-score above **GL-SigRep** even where the coverability is low. The MSE of **GL-conv** follows a similar path with **ML**. Yet, **ML** achieves a lower MSE due to the flexibility in the edge selection process and the corrective term. The F-score of **GL-conv**, on the other hand, is inferior compared to the other methods since it simply merges the topology of the layers without an edge selection process.

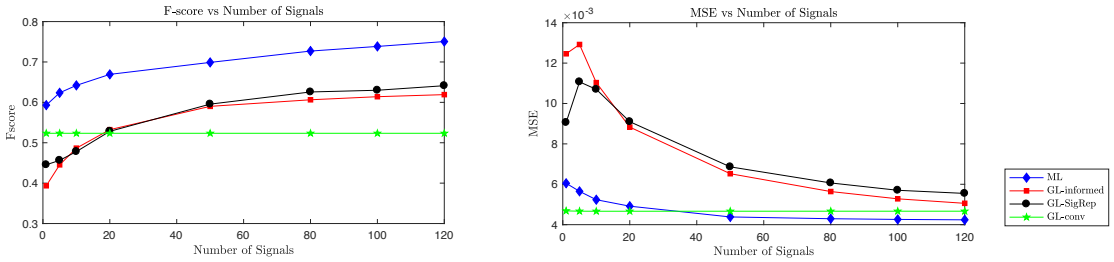


Figure 5.6 – Performance of the algorithms vs number of signals

Effect of signal representation quality. Here, we use a fixed coverability of 0.7 to generate the global graph and the parameter γ for **ML** is set to 100. We first evaluate the global graph recovery of the algorithms by generating different numbers of signals on the global graph. The findings are averaged on 20 different instances of this scenario and plotted in Fig. 5.6. Then,

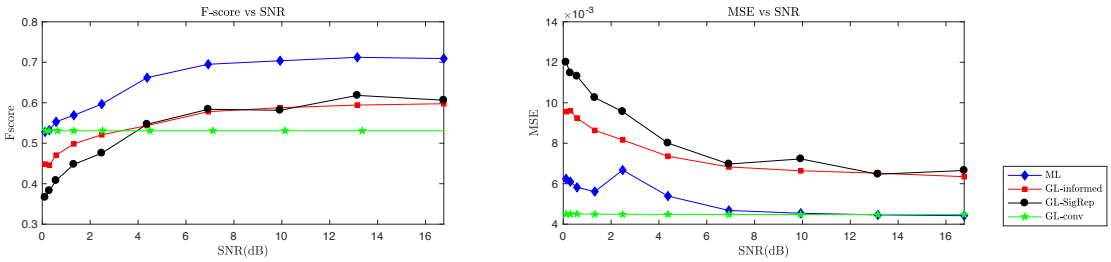


Figure 5.7 – Performance of the algorithms vs signal quality

we measure the performance of the algorithms on signal sets with different SNR values, which is given in Fig. 5.7. To do that, we generate additive noise with normal distribution at different variance values. As expected, all the methods but **GL-conv** achieve better performance as the number of signals increases, or, as the noise power drops. **GL-conv**, on the other hand, is the least affected by the changes in the number of signals. The strictness of the convex combination constraint permits to obtain a similar combination even when there are few signals or noisy signals. Yet, this further prevents enhancing its performance in the high signal representation quality conditions. For instance in Fig. 5.6, **ML** achieves a lower MSE than **GL-conv** when there is a high number of signals. Based on the plots in Fig. 5.5, it is already known that around 70% coverability, **ML** achieves a good performance that is followed by **GL-SigRep** and **GL-informed**. This is also confirmed by the plots in Fig. 5.6 and 5.7. **GL-SigRep** is the method that is the most affected by the signal quality since it is not able to compensate for the lack of observations in the signal set. On the other hand, **ML** is resistant to the change in the signal quality, since it exploits the multi-layer guidance. In addition, **ML** permits flexibility in the learning scheme by adjusting the γ parameter according to the signal quality. For example, in Fig. 5.7, under 2dB SNR, we use $\gamma = 10^7$, so that the learning process relies more on the multi-layer graph representation. Therefore, **ML** manages to perform better than the competitor algorithms in low SNR conditions.

5.3.2 Learning from Meteorological Data

We now present experiments on real datasets and focus first on the meteorological data provided by Swiss Federal Office of Meteorology and Climatology (MeteoSwiss)[II]. The dataset is a compilation of 17 types of measurements including temperature, snowfall, precipitation, humidity, sunshine duration, recorded in weather stations distributed over Switzerland. Monthly normals and yearly averages of the measurements calculated based on the time period 1981-2010 are available at 91 stations. For the stations, we are also provided geographical locations in GPS format and altitude values, i.e., meters above sea level. We use each type of measurement as a different set of observations to feed the graph learning framework. Our goal is to explain the similarity pattern for each type of measurement with the help of geographical

[II] https://www.meteoswiss.admin.ch/home/climate/swiss-climate-in-detail/climate-normals/normal-values-per-measured-parameter.html

location and altitude of the stations.

Multi-Layer Graph Representation. We construct a 2-layer graph representation where the nodes represent the stations, which are connected based on GPS proximity in one layer and based on altitude proximity in the other one. We construct the layers as unweighted graphs by inserting an edge between two stations that have Euclidean distance below a threshold, which is set to an edge sparsity level of 10%. Consequently, each graph layer has approximately the same number of edges so that the edge selection process during mask learning is not biased by any layer. We normalize the adjacency matrices of the layers to fix the volume of the graph layers to the number of nodes, N, which is also used as the value of the parameter Γ in **ML**.

Learning Masks from Different Set of Measurements

We test the mask learning algorithm on different types of observations separately. We use the monthly normal of the measurements as the signal set, which makes the number of signals $K = 12$. Here, the yearly averages are not used for graph learning, instead, they will be used for a visual assessment of the learned graph. We assume that the similarity between the measurement patterns of two stations must be explained either by geographical proximity or elevation similarity. Due to this, we employ **ML** in the reduced version (5.7) to learn a global graph structure with the fully complementary assumption. It is possible to interpret the significance of the geographical location proximity and the altitude proximity in the formation of each type of observation by examining the mask matrices inferred by **ML**.

Table 5.2 – Contribution of layers on the structure of different measurements

Measurement	GPS	Altitude
Temperature	36%	64%
Snowfall (cm)	37%	63%
Humidity	51%	49%
Precipitation (mm)	52%	48%
Cloudy days	65%	35%
Sunshine (h)	54%	46%

In Table 5.2, the percentage of the connections that **ML** draws from the GPS and the altitude layer is given for different types of measurements that are used as signals. To begin with temperature, its structure seems to be highly coherent with the altitude similarity considering the percentage contribution of each layer.

We further check the yearly temperature averages, which is shown in Fig. 5.8. According to that, Bern and Aadorf are the stations providing the most similar average. Indeed, an edge is inferred between them on the global structure of the temperature measurements, and it is extracted from the altitude layer where the two stations are connected within 14m elevation distance. The correlation between temperature measurements and altitude is also noted by the authors in [76]. Similar to temperature, snowfall is also anticipated to be highly

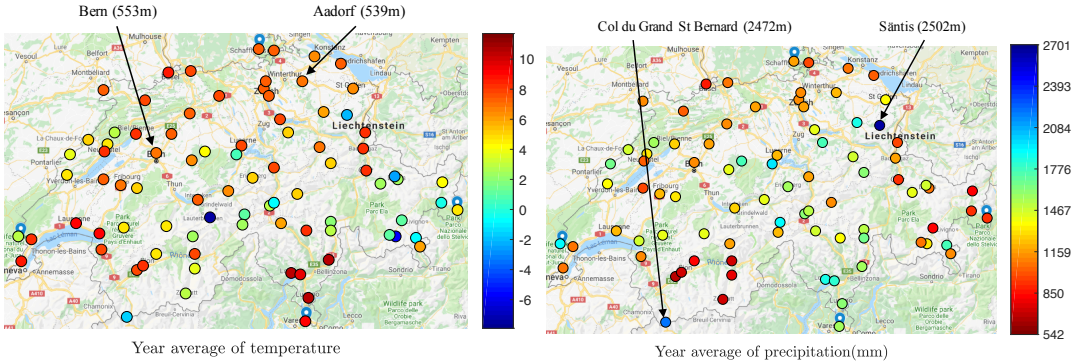


Figure 5.8 – Year average of temperature and precipitation

correlated with the altitude of the stations. This is also what is derived by **ML** which draws more connections from the altitude layer than the GPS layer as given in Table 5.2. The 'cloudy days' measurement, however, is found to be highly coherent with the GPS proximity by drawing 65% of its connections from the GPS layer. Next, humidity, precipitation and sunshine are evenly correlated with both of the GPS and altitude layers, according to Table 5.2. Given the yearly average of precipitation shown in Fig. 5.8, Geneva and Nyon have the closest records. As seen, they are also pretty close on the map and thus their connection on the global graph of precipitation is drawn from the GPS layer. In addition, Fey and Sion are the stations providing the lowest records on average, and their connection is also drawn from the GPS layer. On the other hand, Col du Grand-Saint-Bernard and Säntis display the highest records, and they are connected in the altitude layer with 30m elevation distance between them.

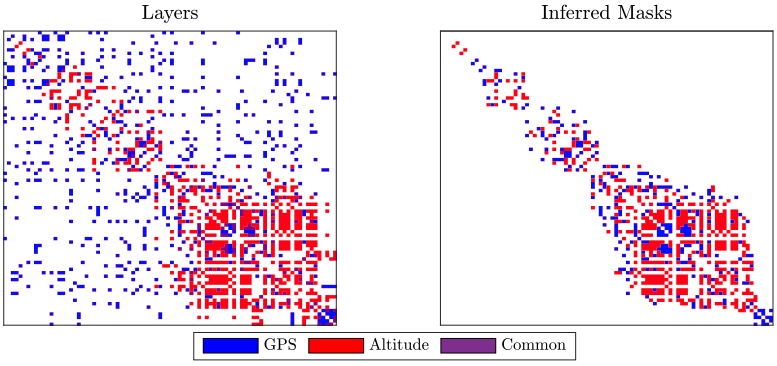


Figure 5.9 – Sparsity pattern of the layers and the masks with respect to year average of temperature

Furthermore, in Fig. 5.9, we visualize the layer adjacency matrices and the inferred mask matrices by sorting the nodes—representing stations—with respect to their yearly average temperature measurements. Recall from Table 5.2 that the altitude layer is found to be dominant for explaining similarities in temperature. This is also evident by the connectivity pattern of the layers, which is shown on the left of Fig. 5.9. The GPS layer connectivity is

distributed broadly whereas the altitude layer connections are gathered around the main diagonal, which contains the edges between the nodes that are similar in yearly average. On the right of Fig. 5.9, we see that inferred mask matrices for both of the layers are organized along the diagonal. This indicates that the algorithm manages to dismiss the connections that are irrelevant to the similarity pattern of temperature, especially on the GPS layer.

Signal Inpainting on the Global Graph

We now prepare a signal inpainting experiment to point out the benefits of learning a proper global graph representation. We consider the monthly normals of the temperature measurements as the signal set. The node set is composed of 86 stations that are providing temperature measurements, i.e., $N = 86$. Then, a graph structure is inferred from those observations using **GL-SigRep**. In addition, by taking the multi-layer graph representation into account, a global graph structure is inferred using **GL-informed**, **GL-conv** and **ML**. During the graph learning process, we train the algorithms by the measurements on 11 months and then try to infer the measurements of the remaining month via inpainting. In the inpainting task, we remove the values of the graph signal to be inpainted—the vector containing the measurements taken on the spared month—on half of the nodes selected randomly. Our aim is to recover the signal values on the whole node space by leveraging the known signal values and the learned graph. We solve the following graph signal inpainting problem [82]:

$$\min_{\mathbf{x}} \quad \|\mathbf{S}\mathbf{x}-\mathbf{y}\|_2^2 + \gamma(\mathbf{x}^\top \mathbf{L}\mathbf{x}), \tag{5.11}$$

which has a closed form solution as:

$$\mathbf{x} = (\mathbf{S}^\top \mathbf{S} + \gamma \mathbf{L})^{-1} \mathbf{S}^\top \mathbf{y}, \tag{5.12}$$

where $\mathbf{y} \in \mathbb{R}^l$ is the vector containing the known signal values by the algorithms, and $\mathbf{x} \in \mathbb{R}^N$ is the vector that contains the recovered signal values on all the nodes. $\mathbf{S} \in \mathbb{R}^{l \times N}$ is a mapping matrix reducing $\mathbf{x}$ to a vector whose entries correspond to the node set with the known signal values. Therefore, $\mathbf{S}^\top \mathbf{S}$ is a diagonal matrix whose non-zero entries correspond to this node set.

We repeat the graph learning and inpainting sequence on 12 instances where the number of signals used in the graph learning part is $K = 11$ and the inpainting is conducted on the values of a different month at each time. We calculate the MSE between the original signal vector and the recovered signal vector. In addition, we compute the *mean absolute percentage error (MAPE)*, which measures the relative absolute error with respect to the original signal magnitudes. We average the performance metrics over 12 instances for each algorithm used in the graph learning part, which is given in Table 5.3. During this experiment, we set $\gamma = 1000$ for **ML** and we normalize the volume of the graph obtained by **GL-conv** to N to provide a fair comparison. Based on the results, **GL-conv** performs poorly compared to other methods, which can be explained by its lack of adaptability to the given signal set. Recall that it finds

Table 5.3 – Signal inpainting performance of the algorithms

	MSE	**MAPE**
GL-SigRep [76]	0.472	12.6%
GL-informed	0.375	13.2%
GL-conv	1.240	14.8%
ML	**0.347**	**10.7**%

a convex combination of the given graph layers in order to fit the smooth signals, which is not very flexible due to the tight search space. **GL-SigRep**, on the other hand, manages to outperform it by learning the structure directly from the signals. **GL-Informed** performs better than **GL-SigRep** in terms of MSE, which indicates that knowing the multi-layer graph representation brings certain advantages. By taking this advantage and coupling it with the flexibility in adapting to the signal set, **ML** leads to a better inpainting performance than the competitors both in terms of MSE and MAPE.

5.3.3 Learning from Social Network Data

Finally, we test our algorithm on the social network dataset[III] provided by [108]. It consists of five kinds of relationship data among 62 employees of the Computer Science Department at Aarhus University (CS-AARHUS), including Facebook, leisure, work, co-authorship and lunch connections. For the experiment, we separate the people into two groups; the first group $\mathcal{A}$ is composed of 32 people having a Facebook account, hence it forms the Facebook network. The second group $\mathcal{B}$ contains any other person eating lunch with anyone in $\mathcal{A}$. The cardinality of $\mathcal{B}$ is 26. We consider a binary matrix $\mathbf{X} \in \mathbb{R}^{32\times 26}$ that stores the lunch records between groups $\mathcal{A}$ and $\mathcal{B}$ as the signal matrix. Our target task is a graph learning problem where we want to discover the lunch connections inside $\mathcal{A}$ by looking at the lunch records between $\mathcal{A}$ and $\mathcal{B}$. For the graph learning problem, we revive the "Friend of my friend is my friend." logic through the smoothness of the signal set. In other words, we assume that two people in $\mathcal{A}$ having lunch with the same person in $\mathcal{B}$ will probably have lunch together. Then, via the mask learning scheme, we exploit the Facebook and work connections among people in $\mathcal{A}$. Hence, the inputs of the mask learning algorithm are (i) the multi-layer graph representation formed by the Facebook and work layers[IV] composing $\mathcal{A}$, which makes the number of nodes in the graph representation $N = 32$, and (ii) the signal set that consists of the lunch records taken on $\mathcal{B}$, which makes the number of signals $K = 26$. Then, the output is the lunch network of $\mathcal{A}$. The number of edges is 124 in Facebook layer and 68 in work layer. The coverability of the union of Facebook and work layers on the ground truth lunch network is 0.84 since the lunch network has 10 connections that do not exist in any of the

[III] http://deim.urv.cat/ alephsys/data.html

[IV] Among the relationships provided by the dataset, we presume that Facebook friendship and colleague relationship within a group could facilitate a substantial prior information in order to predict the lunch activities in that group. If two people have lunch together and they are not colleague then they probably have a social relationship that can be pointed by Facebook connections.

layers. The ground truth lunch network and the one inferred by **ML** are presented in Fig. 5.10 together with a color code for the layers. We compare the performance in terms of the

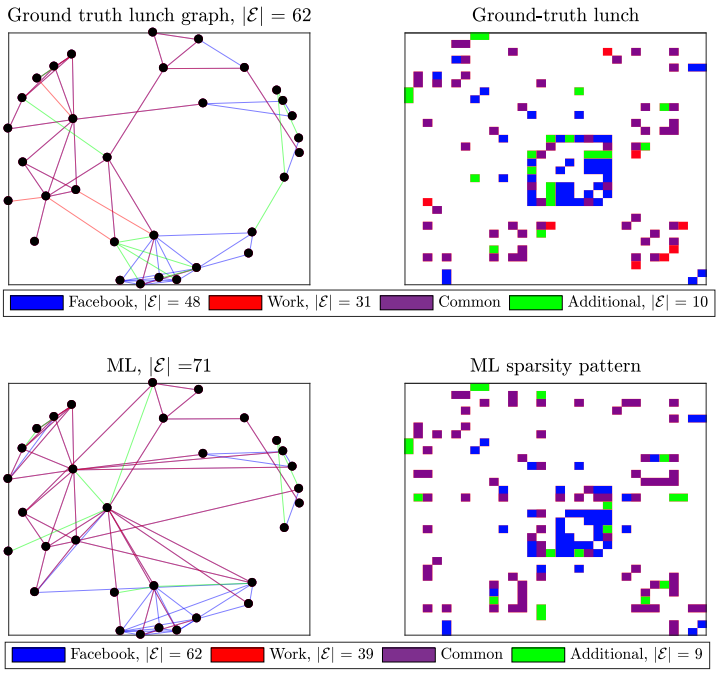


Figure 5.10 – Performance of **ML** ($\gamma = 0.6, \Gamma = 32$) on CS-AARHUS data

retrieval of the lunch network for the following graph learning algorithms: **ML**, **GL-informed**, **GL-SigRep** and the power-sociomatrix that is introduced by [108]. The performance metrics given in Table 5.4 are calculated with respect to the ground truth lunch network and they measure only the link prediction performance since the networks are unweighted. In addition to the precision, recall and F-score, we use the Jaccard index in order to measure a type of similarity between the inferred graph and the ground truth graph. In [108], the Jaccard index is computed for two networks to be compared by the proportion of their intersection to their union and it is 1 when the two have identical topology. Regarding the Jaccard index and the

Table 5.4 – Performance of the methods in recovering the lunch network

		Jaccard	**Recall**	**Precision**	**F-score**
power sociomatrix [108]	{FB}	35%	77%	39%	51%
	{Work}	31%	50%	46%	48%
	{FB,Work}	34%	84%	37%	51%
GL-SigRep [76]		48%	64%	66%	65%
GL-Informed		45%	63%	61%	62%
ML		**58**%	69%	79%	**74**%

F-score, **ML** performs best at the recovery of the lunch network by exploiting the multi-layer representation and the signal set at the same time. With the power-sociomatrix, we obtain all possible combinations of the layers: (i) only the Facebook layer, which is referred to as {FB},

(ii) only the work layer, which is referred to as {Work}, and (iii) the union of the two layers, which is referred to as {FB, Work}. Note that the recall value stated for {FB, Work} also gives the coverability of the multi-layer graph representation, which is computed by dividing the number of lunch connections given by the Facebook or the work layer by the total number of lunch connections. The power-sociomatrix can achieve a limited F-score and Jaccard index since it depends on a simple merging of the two layers without an edge selection process. Then, despite the reasonable coverability rate, **GL-informed** can not reach the performance of **GL-SigRep**, which implies that the signal representation quality is better than the multi-layer representation quality to reach the global graph structure. Yet, when we repeat the experiment with signal sets with a lower number of signals, we observe that **GL-informed** outperforms **GL-SigRep** when the multi-layer graph representation becomes more informative than the signals. The related results are plotted in Fig. 5.11, where we train the algorithms with different numbers of signals, K, at each experiment. Here, the signal set is randomly formed from the lunch records on $\mathcal{B}$ with the corresponding K, and the F-score is averaged over 10 such instances. We employ **ML** in reduced version (5.7) when $K < 10$ so that it depends more on the multi-layer graph representation to compensate for the lack of knowledge from the signal side. This permits **ML** to have the adaptability to different conditions and to outperform the competitor methods as seen in Fig. 5.11.

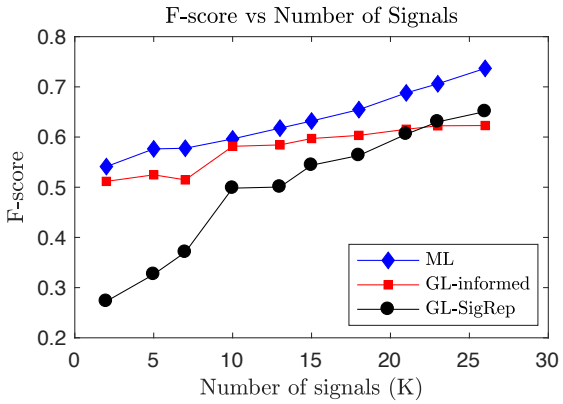


Figure 5.11 – Performance of the graph learning algorithms vs number of signals in lunch data

5.4 Conclusion

In this chapter, we introduced a novel structure inference framework which exploits multi-relational domain knowledge by admitting a multi-layer graph representation of the data space as an input. Our extensive theoretical and experimental analysis shows that the proposed mask learning algorithm is flexible to adjust the inference procedure between the signal representation and the multi-layer graph representation model. This permits adapting to the input data in terms of quality and quantity of the observed signals and reliability of the multi-layer graph representation. The algorithm further outputs a mask combination of the layers indicating relative importance of each layer for the specific structure inference task.

This can be interpreted as revealing the contribution of each relational source of information prioritizing the observed signals.

This study mainly emphasizes the benefit of integrating multi-relational domain knowledge into the structure inference task and hints several research directions for future focus. First, within the scope of structure inference, it is possible to investigate different techniques for combining the multiple types of relationships within data. For instance in the proposed method, the masking strategy can be further specified based on the constraints led by the data domain, such as node-wise masking or locally consistent masking. Moreover, instead of directly imposing a certain signal representation model on the observations, the underlying structure can be learned via a neural generative model such as variational auto-encoder—a similar approach is adopted to learn a directed graph in [133], then they can be further arranged to incorporate multi-relational domain information as in our work. Besides the structure inference task itself, revealing the underlying graph is important for other subsequent machine learning tasks. For instance, given the multi-relational domain information, the computational graph of encoder operation, *e.g.*, graph convolution, can be found specific to a downstream task such as node or graph-level classification or regression.

6 Conclusion

6.1 Summary of Contributions

In this dissertation, we investigated integrating multi-relational domain knowledge with the relational learning models. Then, we show that inclusion of the available multi-relational information about the data domain into the learning framework repays not only with better accuracy but also with the interpretability of the inference task. First, it enhances the performance of the inference task by augmenting the inductive bias with multi-relational prior information, which inherently boosts the relational reasoning capability. Second, such a multi-relational reasoning process reveals the contribution of each relational source of information within data for the inference task of interest.

We can summarize the contributions of each chapter as follows:

In Chapter 2, while giving an overview of the relational representation learning methodologies, we presented a breakdown of the propagation algorithm on a simple weighted graph from a Bayesian perspective.
Local generative model to propagation algorithm: Imposing ℓ_2 sense smoothness prior on connected node representations on the graph, we introduced a local generative model. Following this model, we derived Bayesian estimate of a node's value given its first-hop neighbors. We framed the computation of such a first-order approximate of node's value through neighborhood aggregation. Then, we expressed the propagation algorithm as iterative application of such neighborhood aggregation operations—the development pipeline is depicted in Figure 6.1. We also emphasized that the propagation algorithm iteratively converges to the solution of the graph regularization problem by enlarging the scope of such approximations at each iteration.

In Chapter 3, we studied propagation on multi-relational and directed graphs for node regression. For this purpose, we followed the pipeline in Figure 6.1, which we had analyzed for the standard propagation algorithm previously. Nevertheless, in Chapter 3, we departed from the straightforward ℓ_2 sense smoothness and diversified the prior by considering the multiple

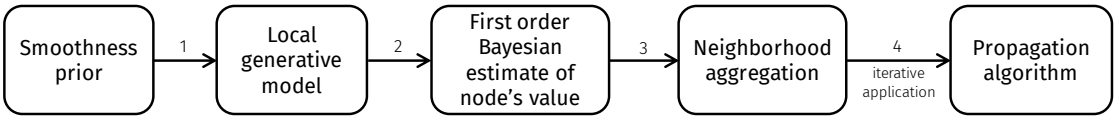


Figure 6.1 – Overview of the pipeline for development of a propagation algorithm

types of directed relationships between data entities. This let us to define a relational local generative model.

Relational local generative model to multi-relational propagation algorithm: We derived the first-order Bayesian estimate of node's value using the relational local generative model. Then, we framed it via an aggregation operation on a multi-relational directed neighborhood. A comparison of the local generative model and the associated neighborhood aggregation operation on simple weighted graphs and multi-relational directed graphs is provided in Table 3.1. Eventually, we proposed an iterative relational neighborhood aggregation scheme to build our multi-relational propagation algorithm, MRP.

In Chapter 4, we studied node attribute completion in knowledge graphs. Regarding both the multi-relational structure of a knowledge graph and the correlation between various types of numerical features possessed by different types of entities, we augmented the relational local generative model. To this end, we introduced heterogeneous message passing functions responsible for information exchange between a source and a target node attribute via multiple types of relations.

Heterogeneous message passing: Based on an iterative heterogeneous message passing scheme, we first proposed a multi-relational attribute propagation algorithm MRAP, where the message passing parameters are estimated in advance to the propagation procedure. Later, we proposed an alternative framework where the parameters and the node attributes are inferred in an end-to-end fashion within a forward-backward learning scheme.

In Chapter 5, we proposed a novel structure inference framework which incorporates the available multi-relational domain knowledge.

Structure inference with multi-relational guidance: We employed multi-layer graphs in order to represent multiple types of relationship between data entities. Then, we adopted smooth signal representation model to impose on a given set of nodal observations and solved its underlying structure as a combination of multi-layer graphs. For this purpose, we introduced a mask combination strategy which relays the relative importance of each layer in terms of structuring the observations.

6.2 Open Research Directions

In this study, we canalized our focus onto certain relational learning tasks that are often overlooked, thus we intend to promote research along these directions. To begin with, in comparison to node and graph classification and link prediction tasks, node regression task is

rather neglected by the recent relational learning studies. For instance, propagation on graphs is usually employed in order to infer categorical node features such as label propagation. However in our study, we introduced propagation frameworks in order to infer continuous node features, and addressed node regression task.

We provided a Bayesian interpretation of the neighborhood aggregation operations accomplished by a propagation algorithm. In our derivation, we assumed certain settings, including uniform prior distribution on the node features. A further analysis can be prompted by integrating a certain prior distribution—if it exists—on the node features rather than assuming it uniform by default, which will further specify the associated neighborhood aggregation operation and the propagation algorithm accordingly. This could be particularly useful for the graph accommodating different types of node features defined on a heterogeneous feature space as in the case of knowledge graphs, since each feature type might rely on a certain prior distribution.

We remind that both of the proposed multi-relational propagation algorithms, MRP and MRAP, are based on the estimation of the propagation parameters in advance via a maximum likelihood estimation. We assigned the uncertainty associated with the prediction of the node's state by its neighbor's as a weight to be taken into account in the neighborhood aggregation. An interesting direction would be to propagate the uncertainties across the graph together with the node features and assert an ultimate uncertainty estimate for the final predictions.

We emphasized that missing facts in a knowledge graph are not only encountered at the edge-level, which has been addressed well by the previous studies, but also at the node-level. This motivated us to develop a method for message passing with incomplete heterogeneous node features, noting that the introduced heterogeneous message passing scheme is open to further improvements. For instance, the proposed end-to-end semi-supervised framework can also be handled from the unified encoder-decoder perspective [134]. It can be then viewed as an auto-encoder which encodes various types node features in terms of relational dependencies between them and then reconstructs them using the learned set of relational rules.

In our framework, we utilized simple linear regression functions for the purpose of message passing, however, non-linear message functions further hint a heterogeneous message passing neural network. For instance, a type of message function can be designed as a multi-layer perceptron. Such a neural learning scheme could be particularly useful in order to infer both numerical and categorical node attributes.

In our message passing scheme, the heterogeneity of the knowledge graph data prompts different types of messages exchanged between the node attributes. These can be considered as set of rules leading the forward propagation. Inferring such rules within a neural-network scheme also motivates a neuro-symbolic learning scheme [35].

Finally, the available multi-relational domain knowledge in many disciplines motivated our focus on integrating it for the structure inference problem. This problem can also be handled

within a Graph auto-encoder scheme [135], encoding and reconstructing the nodal observations in terms of the underlying structure. This might permit releasing a strict statistical model imposed on the observations and rather promote a data-driven model. Alternatively, a prior can be imposed on the underlying graph with an expected network model, *e.g.*, a regular network, a scale-free network, etc. Based on our experimental analysis, we speculate that smooth signal representation model rather prioritizes a regular network model for the underlying graph. The guidance of the prior multi-relational knowledge could be particularly valuable in such a data-driven learning scheme.

An open direction might be also motivated by specifying the structure inference framework for a subsequent machine learning task. The given relational structure of the data may not always constitute the appropriate computational graph for the inference task of interest. In that case, the structure inference framework can be further extended based on our approach.

A Appendix of Chapter 2

A.1 Derivation of Graph Regularization Term as the Quadratic Form of Laplacian

The graph regularization term built with ℓ_2 smoothness prior is introduced as follows:

$$\mathcal{L}_{\text{reg}} = \frac{1}{2} \sum_{i \in \mathcal{V}} \sum_{(i,j) \in \mathcal{E}} \| x_i - x_j \|_2^2. \tag{A.1}$$

Let us open it up by employing the adjacency matrix $\mathbf{A}$:

$$\frac{1}{2} \sum_{i \in \mathcal{V}} \sum_{i \in \mathcal{V}} \mathbf{A}_{ij} (x_i - x_j)^\top (x_i - x_j), \tag{A.2}$$

which can be further unwrapped as follows:

$$\frac{1}{2} \Big(\sum_i \sum_j \mathbf{A}_{ij} x_i^\top (x_i - x_j) + \sum_i \sum_j \mathbf{A}_{ij} x_j^\top (x_j - x_i) \Big). \tag{A.3}$$

Since, the first and second term inside the parenthesis basically express the same sum, we move on as follows:

$$\sum_i \sum_j \mathbf{A}_{ij} x_i^\top (x_i - x_j) \tag{A.4}$$

$$= \sum_i \sum_j \mathbf{A}_{ij} x_i^\top x_i - \sum_i \sum_j \mathbf{A}_{ij} x_i^\top x_j \tag{A.5}$$

$$= \sum_i \mathbf{D}_{ii} x_i^\top x_i - \sum_i x_i^\top (\sum_j \mathbf{A}_{ij} x_j), \tag{A.6}$$

where $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$ is an element of diagonal degree matrix. Then, we re-organize the terms using matrix notation,

$$\text{tr}(\mathbf{X}^\top \mathbf{D} \mathbf{X}) - \text{tr}(\mathbf{X}^\top \mathbf{A} \mathbf{X}) = \text{tr}(\mathbf{X}^\top \mathbf{L} \mathbf{X}). \tag{A.7}$$

A.2 Derivation of Graph Regularization Solution

The problem of graph regularization built with ℓ_2 smoothness prior is stated in (2.3) and the loss can be written as $\mathcal{L} = \mathcal{L}_{\text{reg}} + \|\mathbf{X}-\mathbf{Y}\|_F^2$ where $\mathcal{L}_{\text{reg}} = \text{tr}(\mathbf{X}^\top \mathbf{L}\mathbf{X})$ is introduced as the graph regularization term. The solution is found where the gradient of $\mathcal{L}$ is zero:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{X}} = 2\mathbf{L}\mathbf{X} + 2\gamma(\mathbf{X}-\mathbf{Y}) = \mathbf{O}. \tag{A.8}$$

According to that, the optimal solution is stated in (2.5).

A.3 Iterative Approximation of Graph Regularization Solution

Let us first express the k-th order approximation of the solution in (2.6) using the geometric series expansion (2.7):

$$\mathbf{X}^{(k)} = (1-\xi)\Big(\sum_{t=0}^{k-1}(\xi\mathbf{S})^t\Big)\mathbf{Y}. \tag{A.9}$$

In the same way, we can write and rearrange the $(k+1)$-th order approximation as follows:

$$\begin{aligned}
\mathbf{X}^{(k+1)} &= (1-\xi)\Big(\sum_{t=0}^{k}(\xi\mathbf{S})^t\Big)\mathbf{Y} && \text{(A.10)}\\
&= (1-\xi)\Big(\xi\mathbf{S}\Big(\sum_{t=0}^{k-1}(\xi\mathbf{S})^t\Big) + \mathbf{I}_N\Big)\mathbf{Y} && \text{(A.11)}\\
&= (1-\xi)\xi\mathbf{S}\Big(\sum_{t=0}^{k-1}(\xi\mathbf{S})^t\Big)\mathbf{Y} + (1-\xi)\mathbf{Y} && \text{(A.12)}\\
&= \xi\mathbf{S}\mathbf{X}^{(k)} + (1-\xi)\mathbf{Y} && \text{(A.13)}
\end{aligned}$$

A.4 Negative Log-Likelihood Estimation with the Local Factor Analysis Model

Problem given in (2.15), we use the local factor analysis model stated in (2.11).

$$\underset{x_i}{\text{argmin}} \quad -\sum_{(i,j)\in\mathcal{E}} \log(\text{p}(x_j|x_i)). \tag{A.14}$$

We note that the additive noise in the model is introduced as $\epsilon \sim \mathcal{N}(0, \sigma_{ij}^2\mathbf{I}_d)$. Thus,

$$\text{p}(x_j|x_i) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}}\exp\Big(-\frac{\|x_j-x_i\|_2^2}{2\sigma_{ij}^2}\Big) \tag{A.15}$$

Now, we can minimize the negative log-likelihood along the neighbors as follows:

$$\underset{x_i}{\operatorname{argmin}} \quad \sum_{(i,j)\in\mathcal{E}} \Big(-\log\big(\frac{1}{\sqrt{2\pi\sigma_{ij}^2}}\big) + \frac{\|x_j - x_i\|_2^2}{2\sigma_{ij}^2} \Big) \tag{A.16}$$

Then, we omit the first term inside the sum since it does not depend on the variable we minimize.

B Appendix of Chapter 3

B.1 Gradient of the Loss in Problem (3.3)

$$\frac{\partial \mathcal{L}_i}{\partial x_i} = \sum_{\mathtt{p}\in\mathcal{P}} \left(\sum_{\mathtt{r}(i,j)=\mathtt{p}} \omega_{\mathtt{p}} \Big(x_i - \eta_{\mathtt{p}} x_j - \tau_{\mathtt{p}} \Big) + \sum_{\mathtt{r}(i,j)=\mathtt{p}^{-1}} \omega_{\mathtt{p}} \eta_{\mathtt{p}}^2 \Big(x_i - \frac{x_j}{\eta_{\mathtt{p}}} + \frac{\tau_{\mathtt{p}}}{\eta_{\mathtt{p}}} \Big) \right). \tag{B.1}$$

The solution $\hat{x}_i$ can be obtained by setting the gradient to 0. Thus, the intermediate step to the solution in (3.4) is expressed as

$$\sum_{\mathtt{p}\in\mathcal{P}} \left(\sum_{\mathtt{r}(i,j)=\mathtt{p}} \omega_{\mathtt{p}} \hat{x}_i + \sum_{\mathtt{r}(i,j)=\mathtt{p}^{-1}} \omega_{\mathtt{p}} \eta_{\mathtt{p}}^2 \hat{x}_i \right) = \sum_{\mathtt{p}\in\mathcal{P}} \left(\sum_{\mathtt{r}(i,j)=\mathtt{p}} \omega_{\mathtt{p}} (\eta_{\mathtt{p}} x_j + \tau_{\mathtt{p}}) + \sum_{\mathtt{r}(i,j)=\mathtt{p}^{-1}} \omega_{\mathtt{p}} \eta_{\mathtt{p}}^2 (\frac{x_j}{\eta_{\mathtt{p}}} - \frac{\tau_{\mathtt{p}}}{\eta_{\mathtt{p}}}) \right) \tag{B.2}$$

B.2 Negative Log-Likelihood Estimation of the parameters of the Relational Local Generative Model

Estimation of the model parameters of relation type $\mathtt{p} \in \mathcal{P}$ is realized over the node pairs connected by that relationship as follows:

$$\min_{\tau_{\mathtt{p}}, \eta_{\mathtt{p}} \omega_{\mathtt{p}}} \quad \sum_{i,j\in\mathcal{V} \mid \mathtt{r}(i,j)=\mathtt{p}} \mathcal{L}_{ij}(\tau_{\mathtt{p}}, \eta_{\mathtt{p}} \omega_{\mathtt{p}}) \tag{B.3}$$

where

$$\mathcal{L}_{ij}(\tau_{\mathtt{p}}, \eta_{\mathtt{p}} \omega_{\mathtt{p}}) = -\log\Big(\mathrm{p}\Big((x_i, x_j) \,\Big|\, \tau_{\mathtt{p}}, \eta_{\mathtt{p}} \omega_{\mathtt{p}} \Big) \Big)$$

is the loss originated from negative log-likelihood. Plugging the likelihoods (3.7) in, the solution of the problem can be found by setting the gradient of the sum of the losses over the

node pairs connected by the relation type p to zero:

$$\frac{\sum_{\mathtt{r}(i,j)=\mathtt{p}} \partial \mathcal{L}_{ij}}{\partial \tau_{\mathtt{p}}} = \sum_{\mathtt{r}(i,j)=\mathtt{p}} -\omega_{\mathtt{p}}(x_i - \eta_{\mathtt{p}} x_j - \tau_p) = 0, \tag{B.4}$$

$$\frac{\sum_{\mathtt{r}(i,j)=\mathtt{p}} \partial \mathcal{L}_{ij}}{\partial \eta_{\mathtt{p}}} = \sum_{\mathtt{r}(i,j)=\mathtt{p}} -\omega_{\mathtt{p}} x_j (x_i - \eta_{\mathtt{p}} x_j - \tau_p) = 0, \tag{B.5}$$

$$\frac{\sum_{\mathtt{r}(i,j)=\mathtt{p}} \partial \mathcal{L}_{ij}}{\partial \omega_{\mathtt{p}}} = \sum_{\mathtt{r}(i,j)=\mathtt{p}} -\frac{1}{2\omega_{\mathtt{p}}} + \frac{1}{2}(x_i - \eta_{\mathtt{p}} x_j - \tau_p)^2 = 0. \tag{B.6}$$

Consequently, the set of parameters $\{\tau_{\mathtt{p}}, \eta_{\mathtt{p}} \omega_{\mathtt{p}}\}$ associated with relation p are solved as equivalent to the parameters of a linear regression problem (3.8), (3.9).

Bibliography

[1] Ines Chami, Sami Abu-El-Haija, Bryan Perozzi, Christopher Ré, and Kevin Murphy. "Machine Learning on Graphs: A Model and Comprehensive Taxonomy". In: *CoRR* abs/2005.03675 (2020). arXiv: 2005.03675. URL: https://arxiv.org/abs/2005.03675.

[2] Marinka Zitnik, Francis Nguyen, Bo Wang, Jure Leskovec, Anna Goldenberg, and Michael M Hoffman. "Machine learning for integrating data in biology and medicine: Principles, practice, and opportunities". In: *Information Fusion* 50 (2019), pp. 71–91.

[3] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alan Aspuru-Guzik, and Ryan P Adams. "Convolutional Networks on Graphs for Learning Molecular Fingerprints". In: *Advances in Neural Information Processing Systems* 28 (2015), pp. 2224–2232.

[4] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. "Neural message passing for quantum chemistry". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1263–1272.

[5] Nicola De Cao and Thomas Kipf. "MolGAN: An implicit generative model for small molecular graphs". In: *arXiv preprint arXiv:1805.11973* (2018).

[6] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. "Relational inductive biases, deep learning, and graph networks". In: *arXiv preprint arXiv:1806.01261* (2018).

[7] Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. "Neural relational inference for interacting systems". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 2688–2697.

[8] William L. Hamilton. "Graph Representation Learning". In: *Synthesis Lectures on Artificial Intelligence and Machine Learning* 14.3 (2020), pp. 1–159.

[9] Yao Ma and Jiliang Tang. *Deep Learning on Graphs*. Cambridge University Press, 2020.

[10] Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. "Geometric deep learning: Grids, groups, graphs, geodesics, and gauges". In: *arXiv preprint arXiv:2104.13478* (2021).

[11] Emanuele Cozzo, Guilherme Ferraz de Arruda, Francisco A Rodrigues, and Yamir Moreno. "Multilayer networks: metrics and spectral properties". In: *Interconnected Networks*. Springer, 2016, pp. 17–35.

[12] Stanley Wasserman and Katherine Faust. *Social network analysis: Methods and applications*. Vol. 8. Cambridge university press, 1994.

[13] Barry Bentley, Robyn Branicky, Christopher L Barnes, Yee Lian Chew, Eviatar Yemini, Edward T Bullmore, Petra E Vértes, and William R Schafer. "The multilayer connectome of Caenorhabditis elegans". In: *PLoS computational biology* 12.12 (2016), e1005283.

[14] M Domenico De. "Multilayer modeling and analysis of human brain networks." In: *GigaScience* 6.5 (2017), pp. 1–8.

[15] Michelle M Li, Kexin Huang, and Marinka Zitnik. "Representation learning for networks in biology and medicine: Advancements, challenges, and opportunities". In: *arXiv preprint arXiv:2104.04883* (2021).

[16] Stefano Boccaletti, Ginestra Bianconi, Regino Criado, Charo I Del Genio, Jesús Gómez-Gardenes, Miguel Romance, Irene Sendina-Nadal, Zhen Wang, and Massimiliano Zanin. "The structure and dynamics of multilayer networks". In: *Physics Reports* 544.1 (2014), pp. 1–122.

[17] Alberto Aleta and Yamir Moreno. "Multilayer networks in a nutshell". In: *Annual Review of Condensed Matter Physics* 10 (2019), pp. 45–62.

[18] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P Gleeson, Yamir Moreno, and Mason A Porter. "Multilayer networks". In: *Journal of Complex Networks* 2.3 (2014), pp. 203–271.

[19] Ginestra Bianconi. *Multilayer networks: structure and function*. Oxford university press, 2018.

[20] Carl Yang, Yuxin Xiao, Yu Zhang, Yizhou Sun, and Jiawei Han. "Heterogeneous Network Representation Learning: A Unified Framework with Survey and Benchmark". In: *IEEE Transactions on Knowledge and Data Engineering* (2020).

[21] Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and S Yu Philip. "A survey of heterogeneous information network analysis". In: *IEEE Transactions on Knowledge and Data Engineering* 29.1 (2016), pp. 17–37.

[22] Antoine Bordes, Sumit Chopra, and Jason Weston. "Question Answering with Subgraph Embeddings". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 615–620.

[23] Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J Smola, and Le Song. "Variational reasoning for question answering with knowledge graph". In: *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.

[24] Thomas Gaudelet, Ben Day, A. Jamasb, Jyothish Soman, Cristian Regep, Gertrude Liu, Jeremy B. R. Hayter, Richard J Vickers, Charlie Roberts, Jian Tang, David Roblin, Tom L. Blundell, Michael M. Bronstein, and Jake P. Taylor-King. "Utilizing graph machine learning within drug discovery and development." In: *Briefings in bioinformatics* (2021).

[25] Stephen Bonner, Ian P Barrett, Cheng Ye, Rowan Swiers, Ola Engkvist, and William L Hamilton. "Understanding the Performance of Knowledge Graph Embeddings in Drug Discovery". In: *arXiv preprint arXiv:2105.10488* (2021).

[26] Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. "Product knowledge graph embedding for e-commerce". In: *Proceedings of the 13th International Conference on Web Search and Data Mining*. 2020, pp. 672–680.

[27] Feng-Lin Li, Hehong Chen, Guohai Xu, Tian Qiu, Feng Ji, Ji Zhang, and Haiqing Chen. "AliMeKG: Domain Knowledge Graph Construction and Application in E-commerce". In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2020, pp. 2581–2588.

[28] Antonio Ortega, Pascal Frossard, Jelena Kovačević, José MF Moura, and Pierre Vandergheynst. "Graph signal processing: Overview, challenges, and applications". In: *Proceedings of the IEEE* 106.5 (2018), pp. 808–828.

[29] Xiaowen Dong, Dorina Thanou, Laura Toni, Michael Bronstein, and Pascal Frossard. "Graph signal processing for machine learning: A review and new perspectives". In: *IEEE Signal Processing Magazine* 37.6 (2020), pp. 117–127.

[30] Lise Getoor and Ben Taskar. *Statistical relational learning*. 2007.

[31] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. "A review of relational machine learning for knowledge graphs". In: *Proceedings of the IEEE* 104.1 (2015), pp. 11–33.

[32] Will Hamilton, Payal Bajaj, Marinka Zitnik, Dan Jurafsky, and Jure Leskovec. "Embedding Logical Queries on Knowledge Graphs". In: *Advances in Neural Information Processing Systems* 31 (2018).

[33] Hongyu Ren and Jure Leskovec. "Beta Embeddings for Multi-Hop Logical Reasoning in Knowledge Graphs". In: *Advances in Neural Information Processing Systems* 33 (2020).

[34] Xiaowen Dong, Dorina Thanou, Michael Rabbat, and Pascal Frossard. "Learning graphs from data: A signal representation perspective". In: *IEEE Signal Processing Magazine* 36.3 (2019), pp. 44–63.

[35] L. Lamb, Artur S. d'Avila Garcez, Marco Gori, Marcelo O. R. Prates, Pedro H. C. Avelar, and Moshe Y. Vardi. "Graph Neural Networks Meet Neural-Symbolic Computing: A Survey and Perspective". In: *IJCAI*. 2020.

[36] Eda Bayram. "Propagation on Multi-relational Graphs for Node Regression". In: *arXiv preprint arXiv:2110.08185* (2021).

[37] Eda Bayram, Alberto García-Durán, and Robert West. "Node Attribute Completion in Knowledge Graphs with Multi-Relational Propagation". In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2021, pp. 3590–3594.

[38] Eda Bayram, Dorina Thanou, Elif Vural, and Pascal Frossard. "Mask combination of multi-layer graphs for global structure inference". In: *IEEE Transactions on Signal and Information Processing over Networks* 6 (2020), pp. 394–406.

[39] Dengyong Zhou and Bernhard Schölkopf. "A regularization framework for learning from graph data". In: *ICML 2004 Workshop on Statistical Relational Learning and Its Connections to Other Fields (SRL 2004)*. 2004, pp. 132–137.

[40] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples." In: *Journal of machine learning research* 7.11 (2006).

[41] Jiaxuan You, Rex Ying, and Jure Leskovec. "Position-aware graph neural networks". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 7134–7143.

[42] Mikhail Belkin and Partha Niyogi. "Semi-supervised learning on Riemannian manifolds". In: *Machine learning* 56.1 (2004), pp. 209–239.

[43] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. "Revisiting semi-supervised learning with graph embeddings". In: *International conference on machine learning*. PMLR. 2016, pp. 40–48.

[44] Ulrike Von Luxburg. "A tutorial on spectral clustering". In: *Statistics and computing* 17.4 (2007), pp. 395–416.

[45] Amr Ahmed, Nino Shervashidze, Shravan Narayanamurthy, Vanja Josifovski, and Alexander J Smola. "Distributed large-scale natural graph factorization". In: *Proceedings of the 22nd international conference on World Wide Web*. 2013, pp. 37–48.

[46] Shaosheng Cao, Wei Lu, and Qiongkai Xu. "Grarep: Learning graph representations with global structural information". In: *Proceedings of the 24th ACM international on conference on information and knowledge management*. 2015, pp. 891–900.

[47] Mikhail Belkin and Partha Niyogi. "Laplacian eigenmaps for dimensionality reduction and data representation". In: *Neural computation* 15.6 (2003), pp. 1373–1396.

[48] Dengyong Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, and Bernhard Schölkopf. "Learning with local and global consistency". In: *Advances in neural information processing systems*. 2004, pp. 321–328.

[49] David Kaplan. *Structural equation modeling: Foundations and extensions*. Vol. 10. Sage Publications, 2008.

[50] Sergey Brin and Lawrence Page. "The anatomy of a large-scale hypertextual web search engine". In: *Computer networks and ISDN systems* 30.1-7 (1998), pp. 107–117.

[51] Xiaojin Zhu and Zoubin Ghahramani. “Learning from labeled and unlabeled data with label propagation”. In: *Tech. Rep., Technical Report CMU-CALD-02–107, Carnegie Mellon University* (2002).

[52] Martin Szummer Tommi Jaakkola and Martin Szummer. “Partially labeled classification with Markov random walks”. In: *Advances in neural information processing systems (NIPS)* 14 (2002), pp. 945–952.

[53] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. “Deepwalk: Online learning of social representations”. In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2014, pp. 701–710.

[54] Hongwei Wang and Jure Leskovec. “Unifying Graph Convolutional Neural Networks and Label Propagation”. In: *CoRR* abs/2002.06755 (2020). URL: https://arxiv.org/abs/2002.06755.

[55] Qian Huang, Horace He, Abhay Singh, Ser-Nam Lim, and Austin Benson. “Combining Label Propagation and Simple Models out-performs Graph Neural Networks”. In: *International Conference on Learning Representations*. 2020.

[56] Marco Gori, Gabriele Monfardini, and Franco Scarselli. “A new model for learning in graph domains”. In: *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.* Vol. 2. IEEE. 2005, pp. 729–734.

[57] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. “The graph neural network model”. In: *IEEE transactions on neural networks* 20.1 (2008), pp. 61–80.

[58] Y. Li, Daniel Tarlow, Marc Brockschmidt, and R. Zemel. “Gated Graph Sequence Neural Networks”. In: *International Conference on Learning Representations*. 2015.

[59] Thomas N Kipf and Max Welling. “Semi-supervised classification with graph convolutional networks”. In: *International Conference on Learning Representations*. 2016.

[60] Mikael Henaff, Joan Bruna, and Yann LeCun. “Deep convolutional networks on graph-structured data”. In: *arXiv preprint arXiv:1506.05163* (2015).

[61] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. “Convolutional neural networks on graphs with fast localized spectral filtering”. In: *Advances in neural information processing systems* 29 (2016), pp. 3844–3852.

[62] Vijay Prakash Dwivedi, Chaitanya K Joshi, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. “Benchmarking graph neural networks”. In: *arXiv preprint arXiv:2003.00982* (2020).

[63] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. “Graph Attention Networks”. In: *International Conference on Learning Representations*. 2018.

[64] William L Hamilton, Rex Ying, and Jure Leskovec. “Inductive representation learning on large graphs”. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 2017, pp. 1025–1035.

[65] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. "Representation learning on graphs with jumping knowledge networks". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 5453–5462.

[66] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. "Modeling relational data with graph convolutional networks". In: *European Semantic Web Conference*. Springer. 2018, pp. 593–607.

[67] Thiviyan Thanapalasingam, Lucas van Berkel, Peter Bloem, and Paul Groth. "Relational Graph Convolutional Networks: A Closer Look". In: *arXiv preprint arXiv:2107.10015* (2021).

[68] Dan Busbridge, Dane Sherburn, Pietro Cavallo, and Nils Y Hammerla. "Relational graph attention networks". In: *CoRR* abs/1904.05811 (2019).

[69] Marc Brockschmidt. "GNN-FiLM: Graph Neural Networks with Feature-wise Linear Modulation". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 1144–1152.

[70] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. "Heterogeneous graph attention network". In: *The World Wide Web Conference*. 2019, pp. 2022–2032.

[71] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. "Heterogeneous graph transformer". In: *Proceedings of The Web Conference 2020*. 2020, pp. 2704–2710.

[72] Arthur P Dempster. "Covariance selection". In: *Biometrics* (1972), pp. 157–175.

[73] Nicolai Meinshausen, Peter Bühlmann, et al. "High-dimensional graphs and variable selection with the lasso". In: *Annals of statistics* 34.3 (2006), pp. 1436–1462.

[74] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. "Sparse inverse covariance estimation with the graphical lasso". In: *Biostatistics* 9.3 (2008), pp. 432–441.

[75] Brenden Lake and Joshua Tenenbaum. "Discovering structure by learning sparse graphs". In: (2010).

[76] Xiaowen Dong, Dorina Thanou, Pascal Frossard, and Pierre Vandergheynst. "Learning Laplacian matrix in smooth graph signal representations". In: *IEEE Transactions on Signal Processing* 64.23 (2016), pp. 6160–6173.

[77] Bastien Pasdeloup, Vincent Gripon, Grégoire Mercier, Dominique Pastor, and Michael G Rabbat. "Characterization and inference of graph diffusion processes from observations of stationary signals". In: *IEEE Transactions on Signal and Information Processing over Networks* 4.3 (2017), pp. 481–496.

[78] Santiago Segarra, Antonio G Marques, Gonzalo Mateos, and Alejandro Ribeiro. "Network topology inference from spectral templates". In: *IEEE Transactions on Signal and Information Processing over Networks* 3.3 (2017), pp. 467–483.

[79] Gonzalo Mateos, Santiago Segarra, Antonio G Marques, and Alejandro Ribeiro. "Connecting the dots: Identifying network structure via graph signal processing". In: *IEEE Signal Processing Magazine* 36.3 (2019), pp. 16–43.

[80] Eda Bayram. "Propagation on Multi-relational Graphs for Node Regression". In: *International Conference on Complex Networks and Their Applications*. Springer. 2021.

[81] Siheng Chen, Aliaksei Sandryhaila, George Lederman, Zihao Wang, José MF Moura, Piervincenzo Rizzo, Jacobo Bielak, James H Garrett, and Jelena Kovačevic. "Signal inpainting on graphs via total variation minimization". In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2014, pp. 8267–8271.

[82] Nathanaël Perraudin and Pierre Vandergheynst. "Stationary signal processing on graphs". In: *IEEE Transactions on Signal Processing* 65.13 (2017), pp. 3462–3477.

[83] Felix L Opolka, Aaron Solomon, Cătălina Cangea, Petar Veličković, Pietro Liò, and R Devon Hjelm. "Spatio-temporal deep graph infomax". In: *arXiv preprint arXiv:1904.06316* (2019).

[84] Yajing Wu, Yongqiang Tang, Xuebing Yang, Wensheng Zhang, and Guoping Zhang. "Graph Convolutional Regression Networks for Quantitative Precipitation Estimation". In: *IEEE Geoscience and Remote Sensing Letters* (2020).

[85] Jiehui Deng, Sheng Wan, Xiang Wang, Enmei Tu, Xiaolin Huang, Jie Yang, and Chen Gong. "Edge-Aware Graph Attention Network for Ratio of Edge-User Estimation in Mobile Networks". In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE. 2021, pp. 9988–9995.

[86] Sergei Ivanov and Liudmila Prokhorenkova. "Boost then Convolve: Gradient Boosting Meets Graph Neural Networks". In: *International Conference on Learning Representations*. 2021.

[87] A. C. Rencher and W. Christensen. "Methods of Multivariate Analysis". In: John Wiley & Sons, 2012. Chap. 3, pp. 47–90. DOI: 10.1002/9781118391686.ch3.

[88] Kristina Toutanova and Danqi Chen. "Observed versus latent features for knowledge base and text inference". In: *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*. 2015, pp. 57–66.

[89] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. "Knowledge graph embedding: A survey of approaches and applications". In: *IEEE Transactions on Knowledge and Data Engineering* 29.12 (2017), pp. 2724–2743.

[90] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. "Composition-based Multi-Relational Graph Convolutional Networks". In: *International Conference on Learning Representations*. 2019.

[91] Alberto Garcia-Duran and Mathias Niepert. "Kblrn: End-to-end learning of knowledge base representations with latent, relational, and numerical features". In: *Proceedings of the 34th Conference on Uncertainty in Artificial Intelligence*. 2018.

[92] Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. "End-to-end structure-aware convolutional networks for knowledge base completion". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 3060–3067.

[93] Bhushan Kotnis and Alberto Garcia-Durán. "Learning numerical attributes in knowledge bases". In: *Automated Knowledge Base Construction (AKBC)*. 2018.

[94] Daniel Beck, Gholamreza Haffari, and Trevor Cohn. "Graph-to-Sequence Learning using Gated Graph Neural Networks". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2018, pp. 273–283.

[95] Yao Ma, Suhang Wang, Chara C Aggarwal, Dawei Yin, and Jiliang Tang. "Multi-dimensional graph convolutional networks". In: *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM. 2019, pp. 657–665.

[96] Donghan Yu, Yiming Yang, Ruohong Zhang, and Yuexin Wu. "Generalized Multi-Relational Graph Convolution Network". In: *arXiv preprint arXiv:2006.07331* (2020).

[97] Takuo Hamaguchi, Hidekazu Oiwa, Masashi Shimbo, and Yuji Matsumoto. "Knowledge transfer for out-of-knowledge-base entities: a graph neural network approach". In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 2017, pp. 1802–1808.

[98] Komal Teru, Etienne Denis, and Will Hamilton. "Inductive relation prediction by subgraph reasoning". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 9448–9457.

[99] Daniel Neil, Joss Briody, Alix Lacoste, Aaron Sim, Paidi Creed, and Amir Saffari. "Interpretable graph convolutional neural networks for inference on noisy knowledge graphs". In: *arXiv preprint arXiv:1812.00279* (2018).

[100] Chao Shang, Qinqing Liu, Ko-Shin Chen, Jiangwen Sun, Jin Lu, Jinfeng Yi, and Jinbo Bi. "Edge attention-based multi-relational graph convolutional networks". In: *arXiv preprint arXiv:1802.04944* (2018).

[101] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. "Kgat: Knowledge graph attention network for recommendation". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2019, pp. 950–958.

[102] Xu Chen, Siheng Chen, Jiangchao Yao, Huangjie Zheng, Ya Zhang, and Ivor W Tsang. "Learning on attribute-missing graphs". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

[103] Alberto Garcia-Duran, Sebastijan Dumančić, and Mathias Niepert. "Learning Sequence Encoders for Temporal Knowledge Graph Completion". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 2018, pp. 4816–4821.

[104] Matthias Fey and Jan Eric Lenssen. "Fast graph representation learning with PyTorch Geometric". In: *arXiv preprint arXiv:1903.02428* (2019).

[105] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. "DropEdge: Towards Deep Graph Convolutional Networks on Node Classification". In: *International Conference on Learning Representations*. 2019.

[106] Xiaowen Dong, Pascal Frossard, Pierre Vandergheynst, and Nikolai Nefedov. "Clustering on multi-layer graphs via subspace analysis on Grassmann manifolds". In: *IEEE Transactions on Signal Processing* 62.4 (2013), pp. 905–918.

[107] Renata Khasanova, Xiaowen Dong, and Pascal Frossard. "Multi-modal image retrieval with random walk on multi-layer graphs". In: *2016 IEEE International Symposium on Multimedia (ISM)*. IEEE. 2016, pp. 1–6.

[108] Matteo Magnani, Barbora Micenkova, and Luca Rossi. "Combinatorial analysis of multiple networks". In: *arXiv preprint arXiv:1303.4986* (2013).

[109] Andreas Argyriou, Mark Herbster, and Massimiliano Pontil. "Combining graph Laplacians for semi–supervised learning". In: *Advances in Neural Information Processing Systems*. 2006, pp. 67–74.

[110] Koji Tsuda, Hyunjung Shin, and Bernhard Schölkopf. "Fast protein classification with multiple networks". In: *Bioinformatics* 21.suppl_2 (2005), pp. ii59–ii65.

[111] Pedro Mercado, Francesco Tudisco, and Matthias Hein. "Generalized Matrix Means for Semi-Supervised Learning with Multilayer Graphs". In: *Advances in Neural Information Processing Systems*. 2019, pp. 14848–14857.

[112] Abhishek Kumar, Piyush Rai, and Hal Daume. "Co-regularized multi-view spectral clustering". In: *Advances in Neural Information Processing Systems*. 2011, pp. 1413–1421.

[113] Vikas Sindhwani, Partha Niyogi, and Mikhail Belkin. "A co-regularization approach to semi-supervised learning with multiple views". In: *Proceedings of ICML workshop on learning with multiple views*. Vol. 2005. Citeseer. 2005, pp. 74–79.

[114] Vassilis N Ioannidis, Panagiotis A Traganitis, Yanning Shen, and Georgios B Giannakis. "Kernel-based semi-supervised learning over multilayer graphs". In: *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE. 2018, pp. 1–5.

[115] Jia Chen, Gang Wang, and Georgios B Giannakis. "Multiview canonical correlation analysis over graphs". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019, pp. 2947–2951.

[116] Vassilis N Ioannidis, Antonio G Marques, and Georgios B Giannakis. "A recurrent graph neural network for multi-relational data". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019, pp. 8157–8161.

[117] Kun Zhan, Changqing Zhang, Junpeng Guan, and Junsheng Wang. "Graph learning for multiview clustering". In: *IEEE Transactions on Cybernetics* 99 (2017), pp. 1–9.

[118] Wei Zhou, Hao Wang, and Yan Yang. "Consensus graph learning for incomplete multi-view clustering". In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer. 2019, pp. 529–540.

[119] Sheng Li, Hongfu Liu, Zhiqiang Tao, and Yun Fu. "Multi-view graph learning with adaptive label propagation". In: *2017 IEEE International Conference on Big Data (Big Data)*. IEEE. 2017, pp. 110–115.

[120] Muhammad Raza Khan and Joshua E Blumenstock. "Multi-GCN: Graph Convolutional Networks for Multi-View Networks, with Applications to Global Poverty". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 606–613.

[121] Hilmi E Egilmez, Eduardo Pavez, and Antonio Ortega. "Graph learning from data under Laplacian and structural constraints". In: *IEEE Journal of Selected Topics in Signal Processing* 11.6 (2017), pp. 825–841.

[122] Vassilis Kalofolias and Nathanaël Perraudin. "Large Scale Graph Learning From Smooth Signals". In: *International Conference on Learning Representations*. 2018.

[123] Santiago Segarra, Yuhao Wangt, Caroline Uhler, and Antonio G Marques. "Joint inference of networks from stationary graph signals". In: *2017 51st Asilomar Conference on Signals, Systems, and Computers*. IEEE. 2017, pp. 975–979.

[124] Yuhao Wang, Santiago Segarra, and Caroline Uhler. "High-dimensional joint estimation of multiple directed Gaussian graphical models". In: *Electronic Journal of Statistics* 14 (2020), pp. 2439–2483.

[125] Hermina Petric Maretic and Pascal Frossard. "Graph Laplacian mixture model". In: *IEEE Transactions on Signal and Information Processing over Networks* 6 (2020), pp. 261–270.

[126] Hermina Petric Maretic, Mireille El Gheche, and Pascal Frossard. "Graph heat mixture model learning". In: *2018 52nd Asilomar Conference on Signals, Systems, and Computers*. IEEE. 2018, pp. 1003–1007.

[127] Kim-Chuan Toh, Michael J Todd, and Reha H Tütüncü. "SDPT3—a MATLAB software package for semidefinite programming, version 1.3". In: *Optimization methods and software* 11.1-4 (1999), pp. 545–581.

[128] Hisayuki Tsukuma and Yoshihiko Konno. "On improved estimation of normal precision matrix and discriminant coefficients". In: *Journal of multivariate Analysis* 97.7 (2006), pp. 1477–1500.

[129] T Tony Cai, Weidong Liu, Harrison H Zhou, et al. "Estimating sparse precision matrix: Optimal rates of convergence and adaptive estimation". In: *The Annals of Statistics* 44.2 (2016), pp. 455–488.

[130] Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze. "Introduction to information retrieval". In: *Natural Language Engineering* 16.1 (2010), pp. 100–103.

[131] Michael Grant and Stephen Boyd. *CVX: Matlab software for disciplined convex programming, version 2.1*. 2014.

[132] MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 8.1*. 2017. URL: http://docs.mosek.com/8.1/toolbox/index.html.

[133] Yue Yu, Jie Chen, Tian Gao, and Mo Yu. "Dag-gnn: Dag structure learning with graph neural networks". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 7154–7163.

[134] William L. Hamilton, Rex Ying, and Jure Leskovec. "Representation Learning on Graphs: Methods and Applications". In: *IEEE Data Eng. Bull.* 40.3 (2017), pp. 52–74.

[135] Thomas N Kipf and Max Welling. "Variational Graph Auto-Encoders". In: *stat* 1050 (2016), p. 21.

Eda Bayram

Contact Information	E-mail: bayrameda.ee@gmail.com Homepage: bayrameda.github.io

Qualifications And Interests	**Research Interests:** Knowledge Representation and Reasoning, Relational Machine Learning, Statistical Inference **Background:** Linear Algebra, Probability, Optimization, Signal Processing, Network Data Science **Coding:** Python, experience with ML libraries e.g., PyTorch , C/C++ and Matlab **Languages:** Turkish (native), English (fluent), French (B1), German (A2)

Education

Sep 2021 - Apr 2017 — PhD, Electrical and Electronics Engineering
École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

- Doctoral assistant involving in teaching and research in *Signal Processing*, *Network Data Science*, *Graph Representation Learning*, *Knowledge Graphs*
- Thesis study: Representation Learning on Multi-relational Data

Advisor: Prof. Pierre Vandergheynst

Feb 2017 - Sep 2014 — MSc, Electrical and Electronics Engineering (Honor Student, GPA: 3.71/4.00)
Middle East Technical University (METU), Turkey

- Specialization on adaptive signal processing with courseworks on linear algebra, optimization, machine learning, computer vision and image processing
- Thesis study on exploitation of spectral graph theory and graph signal processing frameworks for the analysis of 3D LiDAR point clouds under co-supervision of Prof. Aydın Alatan and Elif Vural.

Jun 2013 - Sep 2009 — BSc, Electrical and Electronics Engineering (Honor Student, GPA: 3.38/4.00)
Middle East Technical University (METU), Turkey

- Senior year specialization on telecommunications and signal processing
- Graduation project: Interior route-finding, wearable assistive device for visual-defective people

Work and Research Experience

Jan 2017 - May 2015 — System and Design Engineer
Aselsan SST (R&D Defense Industry), Turkey

- Research and development of target tracking algorithms for day-TV and thermal camera systems.

Sep 2015 - Jun 2015 — Research Intern
Signal Processing Laboratory, EPFL, Switzerland

- Spectral graph-based motion estimation for omnidirectional videos

Apr 2015 - Jul 2013 — Software Engineer
Aselsan MGEO (R&D Defense Industry), Turkey

- Software design and development for the communication of the peripheral modules in embedded systems.

Jun 2013 - Nov 2012 — Part-time Software Engineer
Aselsan SST (R&D Defense Industry), Turkey

Teaching Activity

Teaching Assistantship

Fall 2017-18	*Digital Signal Processing*, Bachelor course at EPFL Electrical Engineering
Fall 2018-19	*Network Tour of Data Science*, Master course at EPFL Electrical Engineering
Fall 2019-20	*Network Tour of Data Science*, Master course at EPFL Electrical Engineering

Supervision in Semester Projects of Master Students

Fall 2017-18	*Building Extraction on Aerial LiDAR Point Clouds using Spectral Graph Features*
Fall 2017-18	*Semi-Supervised Learning and Inpainting on Multi-Layer Graphs*
Spring 2018-19	*Spectral Graph Filter Learning for Point Labeling in Airborne LiDAR Data*
Fall 2019-20	*Informed Source Separation for Multi-Modal Graph Signals*
Spring 2020-21	*Completion of Heterogeneous Node Features with Message Passing Neural Net*

Formation in Teaching

- Course work in *Teaching Science & Engineering*
- Participation in Workshop for *Presenting: Explaining Scientific Concepts* by EPFL Teaching Center
- Participation in Workshop *Teaching Toolkit for Projects* by EPFL Teaching Center

Academic Cohorts

Talks and Conference Participation

- Participation and presenter in *IEEE SPS/EURASIP Summer School in Network and Data-Driven Learning Fundamentals*, May 2019, Lecce, Itay
- Seminar on *Structure Inference* @ METU, OGAM, Jan 2020, Ankara, Turkey
- Presenter in *2020 IMPRS-IS PhD Symposium* @ The International Max Planck Research School for Intelligent Systems, Tubingen, Germany
- Lightning Talk @ *WiDS Cambridge 2021* virtual conference
- Participation and presenter in *IJCLR 2021 Workshop: Statistical Relational AI (StarAI)*

Event Organization

- Organization team member @ *3rd Graph Signal Processing Workshop – GSP'18* in Lausanne, Switzerland
- Volunteer in the organization of *SIGKDD'20*, virtual conference
- Volunteer in the organization and presenter in *WiML 2020* @NeurIPS virtual conference
- Program comittee member of *Workshop on Graph Neural Networks and Systems – GNNSys'21* @ MLSys virtual conference

Others

- Reviewer for IEEE-TSIPN *Transactions on Signal and Information Processing over Networks*
- Research Fellowship in *DAAD AINet 2021*, German Academic Exchange Service (DAAD)
- Participation in the mentoring program organized by *Fix the Leaky Pipeline* @ ETH domain

Publications

- Eda Bayram, "Propagation on Multi-relational Graphs for Node Regression", *International Conference on Complex Networks and Their Applications, 2021.* Springer
- Eda Bayram, Alberto Garcia Duran and Robert West, "Node Attribute Completion on Knowledge Graphs with Multi-Relational Propagation", *ICASSP - IEEE International Conference on Acoustics, Speech and Signal Processing, 2021*
- Eda Bayram, Dorina Thanou, Elif Vural and Pascal Frossard, "Mask Combination of Multilayer Graphs for Global Structure Inference", *IEEE Transactions on Signal and Information Processing over Networks 6 : 394-406*, 2020
- Eda Bayram, Pascal Frossard, Elif Vural and Aydin Alatan, "Analysis of airborne LiDAR point clouds with spectral graph filtering", *IEEE Geoscience and Remote Sensing Letters 15.8 : 1284-1288.*, 2018
- Eda Bayram, Elif Vural, and Aydin Alatan. "A graph signal filtering-based approach for detection of different edge types on airborne lidar data" *Lidar Technologies, Techniques, and Measurements for Atmospheric Remote Sensing XIII. Vol. 10429. International Society for Optics and Photonics*, 2017.
- Eda Bayram, "Spectral Graph Based Approach for Analysis of 3D LiDAR Point Clouds" *Master Thesis, METU*, Ankara, 2017.